Kentico CMS 5.5 R2 Developer's Guide

Table of Contents

Part I

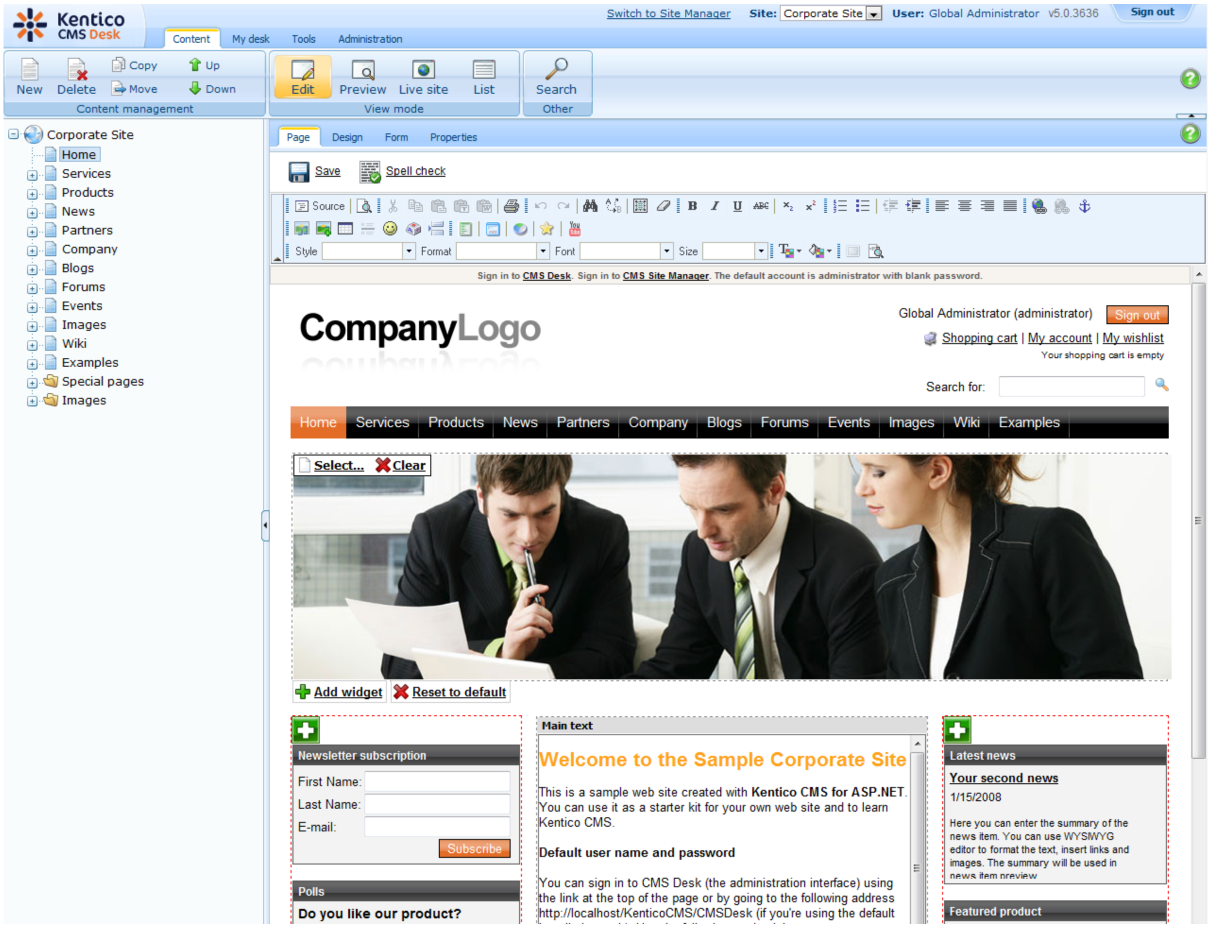
Introduction

1 Introduction

1.1 Introduction

Thank you for using Kentico CMS, the content management system for ASP.NET developers. Kentico CMS 5.5 R2 empowers developers in creation of dynamic websites. It provides rich functionality out of the box:

- Flexible content repository for storing structured content with versioning and workflow.
- Powerful portal engine for assembling web pages from web parts.
- A wide range of built-in web parts and web controls.
- Export and import for easy deployment.
- Granular security model.
- Support for multiple-site configuration.
- Support for multiple-languages on a single website.
- Many built-in modules:
  - E-commerce
  - On-line forms
  - Forums
  - Image gallery
  - Media libraries
  - Content staging
  - Content rating
  - Blogs
  - Polls
  - Reporting
  - Web Analytics
  - Time zones
  - Booking system
  - User contributions (Wiki)
  - and others



Kentico CMS enables a **high level of flexibility**, **customization** and **integration** – you can create:

- custom page templates
- custom web parts
- custom document types
- custom workflow schemas
- custom modules

Kentico CMS object model and **open, well-documented application programming interface (API)** allow you to enhance the system and script any system feature by your .NET code. You can also write custom handlers to add your own actions when a system event (such as creation of a new document) occurs.

More product information

You can download Kentico CMS, find additional information, make the purchase and get support at http://www.kentico.com

1.2 About this guide

Who is this guide for?

Kentico CMS Developer's Guide contains reference information for Kentico CMS developers and system administrators.

It is expected that you are familiar with the basics of Kentico CMS explained in Kentico CMS Tutorial (or Kentico CMS Tutorial ASPX if you are developing in the ASPX page templates mode). It is also expected that you have a certain degree of experience in ASP.NET web development and C# programming to be able to fully understand all articles in this guide.

What information can I find in the guide?

This guide should provide comprehensive reference for website development and administration in Kentico CMS. It provides description of the system's functionality and step-by-step examples of typical development and administration tasks.

How is the guide structured?

The guide is divided into several logical chapters:

- Kentico CMS overview - basic information about the system and intended for a first-time user
- Installation and deployment - covers the installation process, additional configuration tasks that might be needed after installation and a troubleshooting section with solutions to usual installation issues
- Content management - information on how to manage the content of your website; it explains how pages can be edited, how the WYSIWYG editor can be used, how files can be managed, etc.
- Managing sites - management of sites, i.e. creation of a new site, running and stopping sites, import and export of sites and objects, license management, etc.
- website settings - overview chapter with links to other sections of the guide where various website settings are explained in different contexts
- Development - provides information on various topics related to website development and system administration
- Modules - reference on in-box modules in Kentico CMS
- API programming and Kentico CMS internals - examples of the usage of Kentico CMS API and explanation of the internals of the system

Valuable reference can also be found in the four appendices:

- Appendix A - Macro expressions - explains the possibilities of using macro expressions in Kentico CMS
- Appendix B - Path expressions - explains how path expressions can be entered in various parts of the system
- Appendix C - Web.config paramenters - list of keys which can be added to the *web.config* file of your site to perform additional low-level settings

1.3 Where do I get further help?

This guide is not the only reference available. All documentation can be get in various formats at devnet.kentico.com/documentation.aspx.

If you can't find the information that you are looking for, please visit devnet.kentico.com. This is the official portal for Kentico CMS developers where you can find:

- Blogs
- Discussion forums
- Knowledge base
- FAQs

Paid license owners can also contact our highly-responsive support team, please visit www.kentico.com/support.aspx to learn more details.

You are also welcome to send us feedback on particular topics in this guide. You can do it by clicking the **Mail us feedback on this topic!** button at the top right corner of the HTML and CHM versions.

1.4 Version history

Listing of all new features, minor improvements and bug fixes in each released version of Kentico CMS can be found at http://www.kentico.com/Download/Version_History.aspx.

Part II

Kentico CMS overview

2 Kentico CMS overview

2.1 What are the benefits of Kentico CMS?

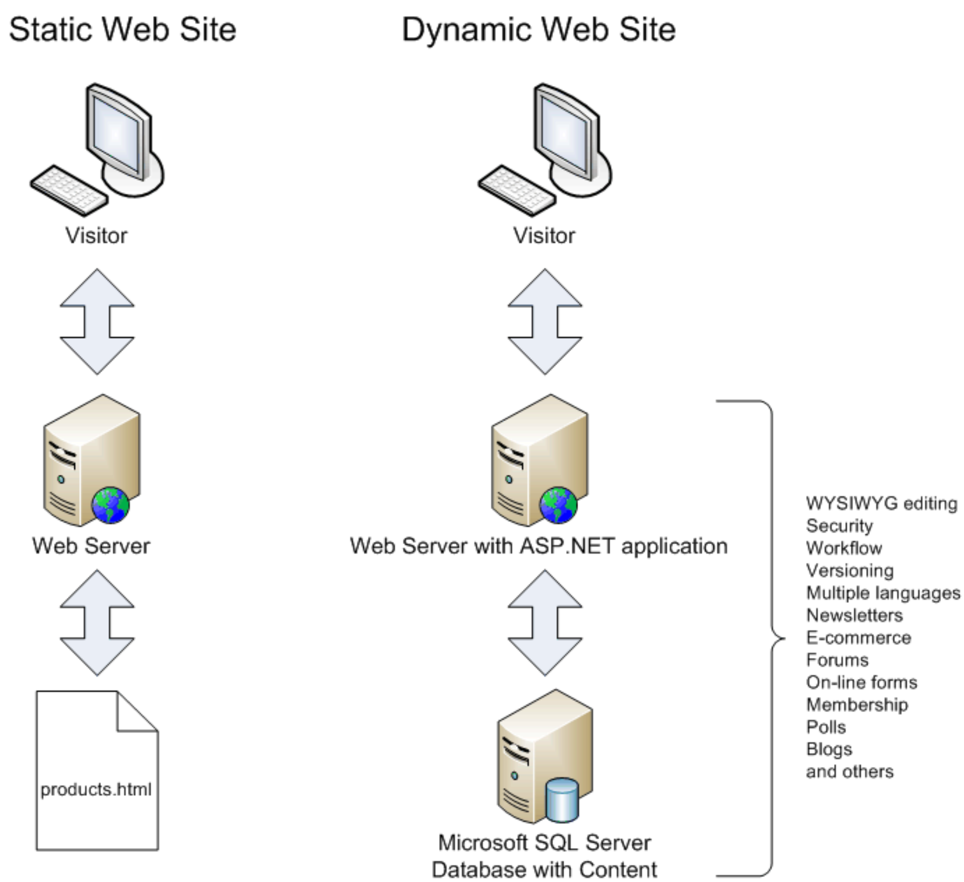
Before you dive into the details, it may be useful to understand the big picture. This chapter explains how the system works, describes its architecture and answers the most common questions you may have.

Is Kentico CMS a standard ASP.NET application?

Yes, Kentico CMS is a standard ASP.NET 2.0 application written in C#. It uses only standard functionality of .NET managed code which means you can use it on basically any ASP.NET 2.0-enabled web server without complicated configuration.

How does the CMS work?

The CMS allows you to manage content of dynamic websites. Unlike static websites that uses static HTML files stored on the disk, a dynamic website displays content from the database. Kentico CMS provides both content storage and all surrounding infrastructure to manage the content and display it on the website. Kentico CMS doesn't pre-render static HTML pages; instead, it renders the content in real time, when it's requested by the visitor.



The main advantages of dynamic website with content management system include:

- **Easy content editing** through a WYSIWYG, browser-based interface for non-technical users
- **Content re-use** - you can display the same structured content in various ways while managing the data at one place
- **Multi-user environment** - website management is not limited to a single web developer
- **Additional functionality and applications**, such as Newsletters, E-commerce, Forums, etc.

Why should I use Kentico CMS and what benefits does it bring to me?

Kentico CMS simplifies the development of dynamic websites. Instead of developing the whole infrastructure for editing, you can utilize the flexible content management framework of Kentico CMS and focus on the site-specific functionality and design. If you consider how much time you would spend only by developing the security system, there's no doubt you should use an existing framework.

With Kentico CMS, you:

- save time and money by developing the dynamic website faster
- focus on the client's business needs instead of core infrastructure
- provide your client with additional functionality, such as Newsletters, Forums, and others that would be difficult and expensive to develop

Is it flexible enough for my needs?

Well, now you may think "If I develop the website from scratch I can create the system and enhance it at any time as my client requests." Yes, you're right, but you can do the same with Kentico CMS. Kentico CMS has been used for hundreds of websites worldwide and it was designed to fit various needs of web developers and their clients. Beside, if you need to add extra functionality, you can:

- create your own modules
- add your own code to the pages
- modify default system behavior using custom handlers and providers
- customize the core engine of the system (if you purchase the source code version)

Kentico CMS was designed for the needs of web developers and their clients. You can be confident that you can **implement basically any website structure, navigation, graphic design and integrate custom functionality**.

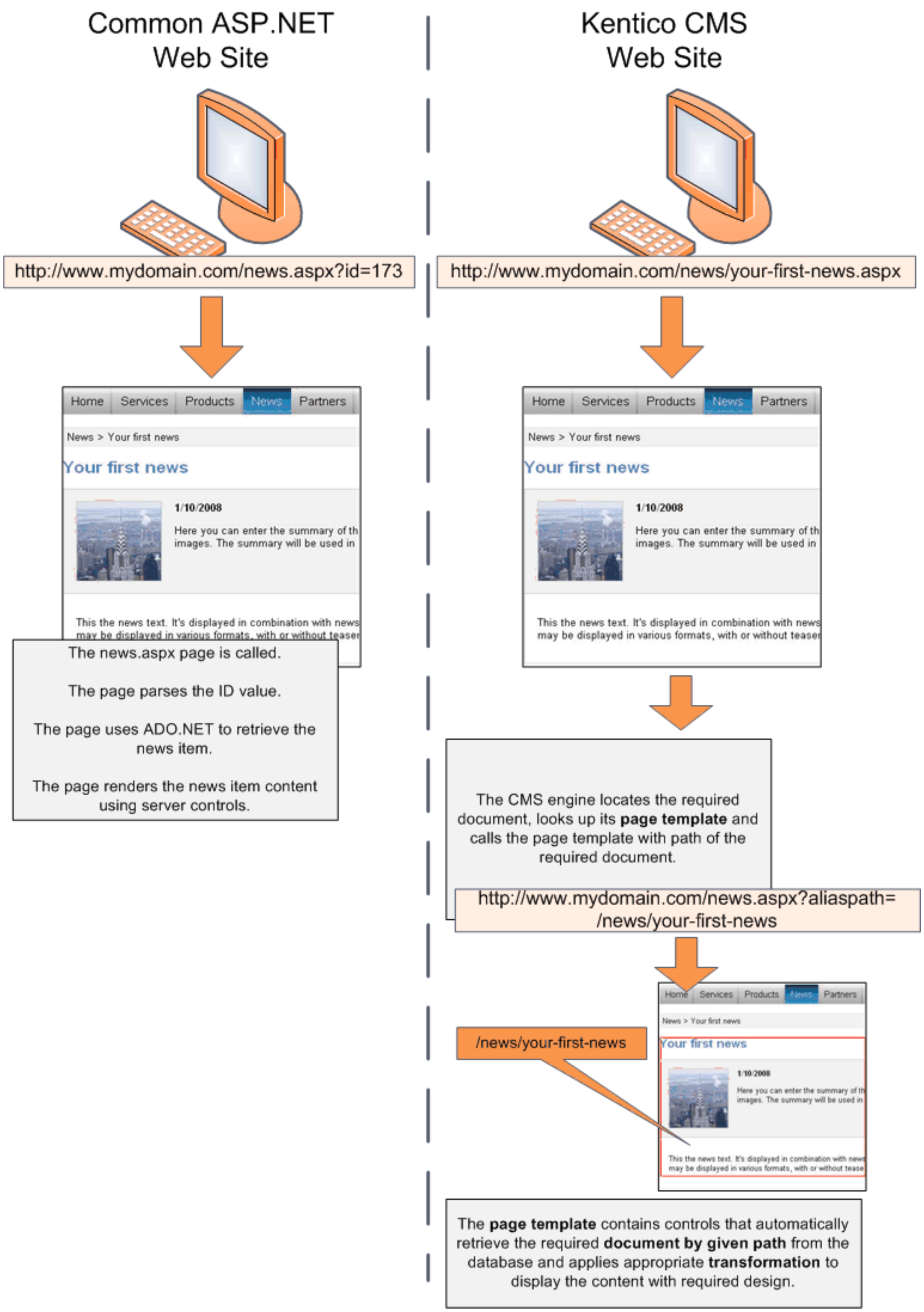
Don't take our word for it

Please visit the Kentico CMS Showcase at http://www.kentico.com/Showcase.aspx for reference websites, clients and testimonials.

2.2 How does it really work?

If you're familiar with dynamic websites, you may want to know what the difference between a CMS system and a common ASP.NET website is. Technically, Kentico CMS is just another ASP.NET web application. It's advantage is that it provides a ready-to use framework for all common tasks.

Here's a comparison of page processing in a typical ASP.NET and Kentico CMS:



But it looks more complex!

Yes, the CMS system is more complex to make your job easier. In Kentico CMS, you do not need to develop complex pages, write ADO.NET code or SQL queries.

The CMS does much of that for you and you usually only drag-and-drop controls or web parts and set their properties. Then, you write .NET code only if you need to add additional business logic or functionality that is not supported out-of-the-box.

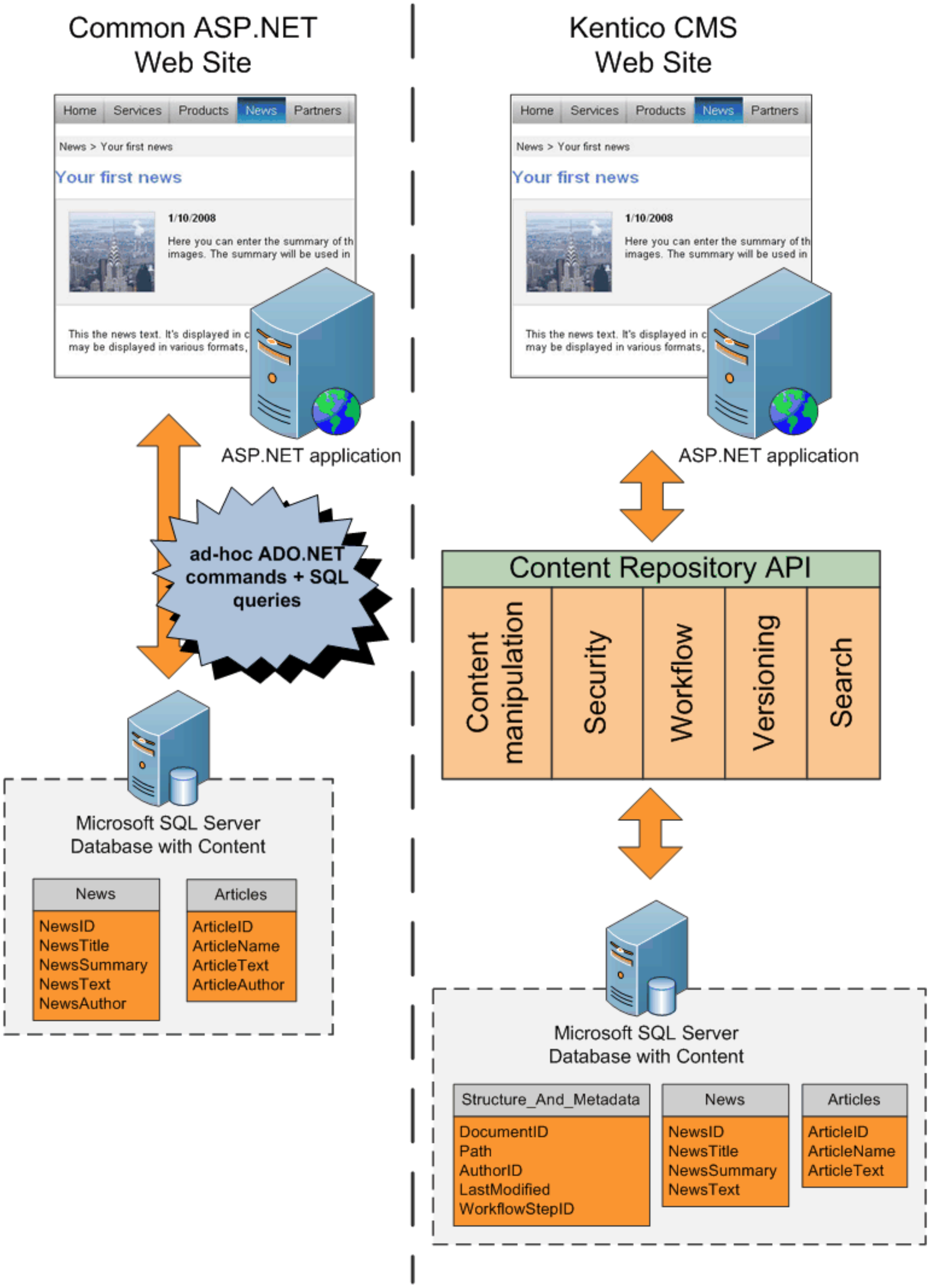
What else does the CMS do for me?

Kentico CMS provides a powerful content repository for your web content. Read the next chapter to learn more.

2.3 Where is the content stored?

Kentico CMS provides a content management sub-system, also known as **content repository**, that allows you to organize website structure and content. Moreover, it provides **a layer of security, workflow, versioning, search** and other services. All types of content can be retrieved and modified through a single **Application Programming Interface (API)**.

The following figure explains the difference between common data access approach and a content repository of Kentico CMS:



As you can see, a common approach to building dynamic websites is to write code for every page and every content type. It means that you need to write similar ADO.NET commands and SQL queries over and over again. With unified content repository, you use a complete set of API methods that allow you to save and retrieve any content type using methods someone wrote for you.

It greatly simplifies the management and retrieval of content since:

- you do not need to write your own methods, **you only call API or use built-in controls**
- **same rules and mechanisms can be applied to any content type**, without writing additional code for every new type

An important part of content repository is **metadata**. The metadata includes:

- content organization into a tree hierarchy that also represents the site map
- information about content authors and modifications
- workflow-related details, such as current workflow step
- content expiration
- permissions for the given document

In the example above, you can see that with classic approach, both News and Articles tables contain the attribute Author. In the content repository, the author is stored in shared metadata for all documents, regardless of their type.

> **Please note**
>
> All metafiles and attachments are stored under a folder specified in **Site Manager -> Settings -> Files -> Files folder**. If this path is not set:
>
> - metafiles of objects not connected with a particular site (global objects) are stored under *<web root>/CMSFiles*
> - metafiles of objects connected with a particular site are stored under *<web root>/<site name>/metafiles*
> - attachments (always connected with a particular site) are stored under *<web root>/site name>/files*

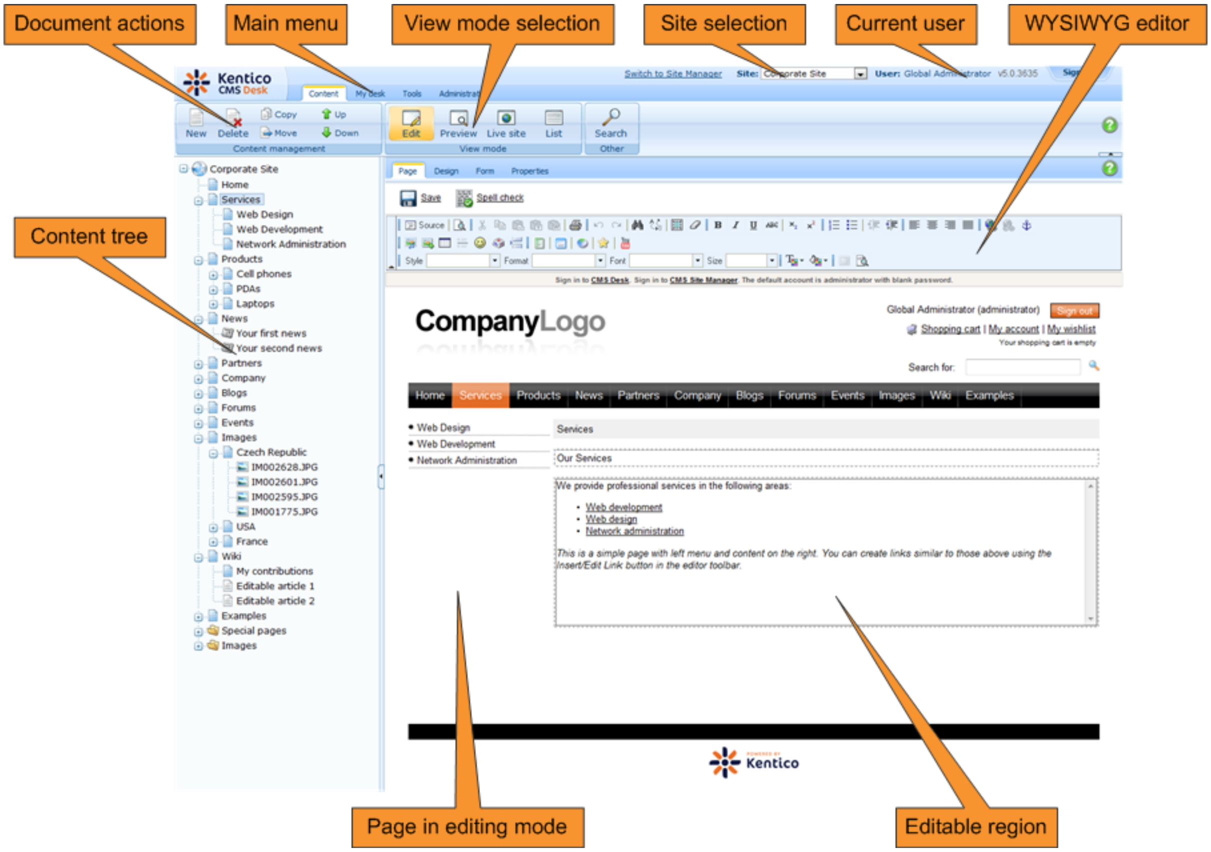
As you can see, the **content repository represents a systematic approach to content storage, manipulation and security**.

2.4 How do I edit content?

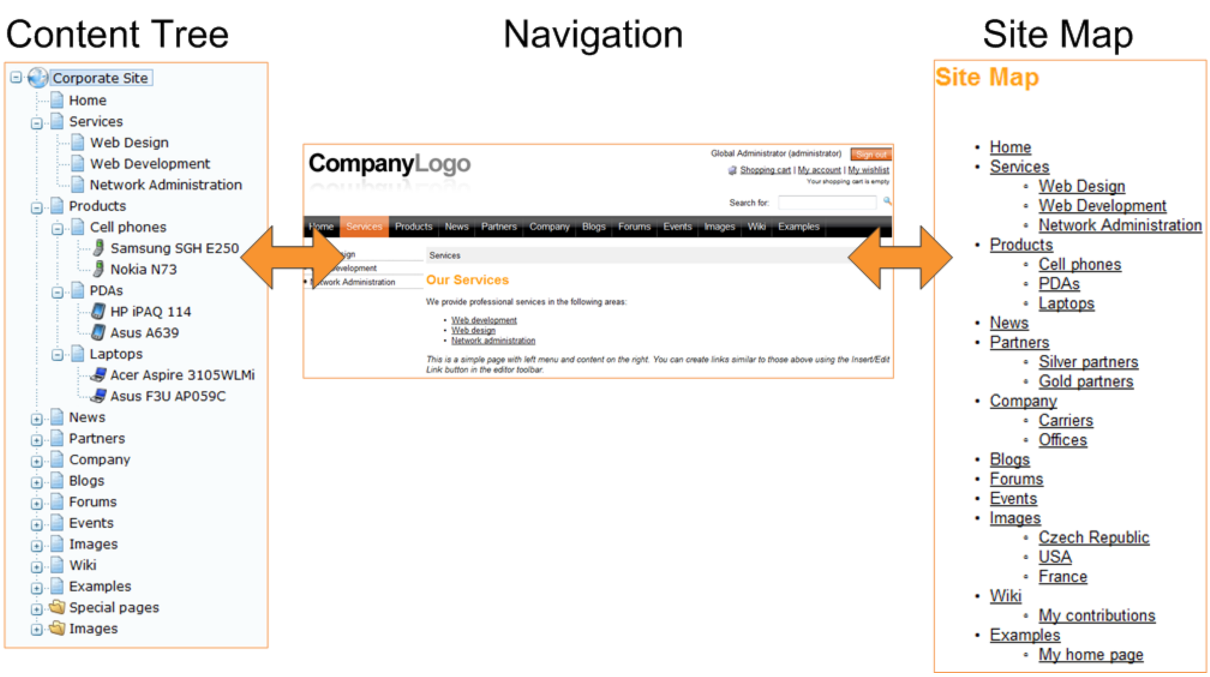
Until now, we were talking about the architecture and how the system works. But how do I edit the content? After all, it's the most important feature of a content management system.

Kentico CMS comes with browser-based administration interface. It's divided into two parts:
- **CMS Desk** (typically http://www.example.com/cmsdesk) - the user interface for content editors
- **CMS Site Manager** (typically http://www.example.com/cmssitemanager) - the user interface for administrators and web developers

The following figure shows the CMS Desk user interface.



The left **content tree** allows you to browse the content and choose the document you want to edit. The content tree also represents a site map of the site and it's used for rendering navigation. In the following figure, you can see how the content tree, navigation and site map fit together:



The **action menu** allows you to create, delete, copy, move and order documents.

The **view mode** allows you to switch between the following views:

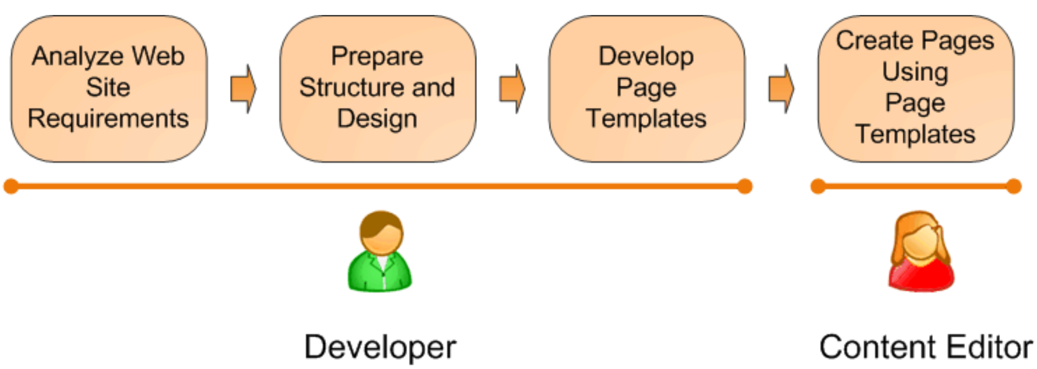
- **Edit** - editing mode
- **Preview** - preview mode. It displays the current version of the page before it's published (if you're using workflow). It also displays the content without using caching, which allows you to preview the content even if the live website displays the cached version.
- **Live site** - live site view
- **List** - displays a list of child documents under the currently selected document. It can be used for delete of multiple documents at the same time and it's useful if there are too many documents in the given section and cannot be browsed comfortably in the content tree.

The **editing area** allows you to edit the content and metadata of the document you selected in the content tree. You can choose from the following tabs/editing modes:

- **Page** - here you can edit the content of editable regions that are used for unstructured content. Besides, any document can have also structured content that can be edited in the editing form on the Form tab.
- **Design** - here you can modify the page layout and web parts (this applies to portal engine development that will be described later in this guide). This tab is only available for global administrators (developers).
- **Form** - here you can modify the structured data, such as news title, news summary, release date, etc.
- **Product** - here you can modify the product specification if the given document represents a product that can be added to the shopping cart (if you're using the e-commerce module).
- **Properties** - here you can modify various settings, permissions, metadata and design settings of the document.

2.5 How do I develop a website?

Now that you know how to edit the content, you may want to know how to develop the website and manage the design. Although these topics will be described later in this guide, let's have take a quick overview of the development process:



This figure shows how you develop the website and how the roles are split between developers and editors. A typical development process consists of the following steps:

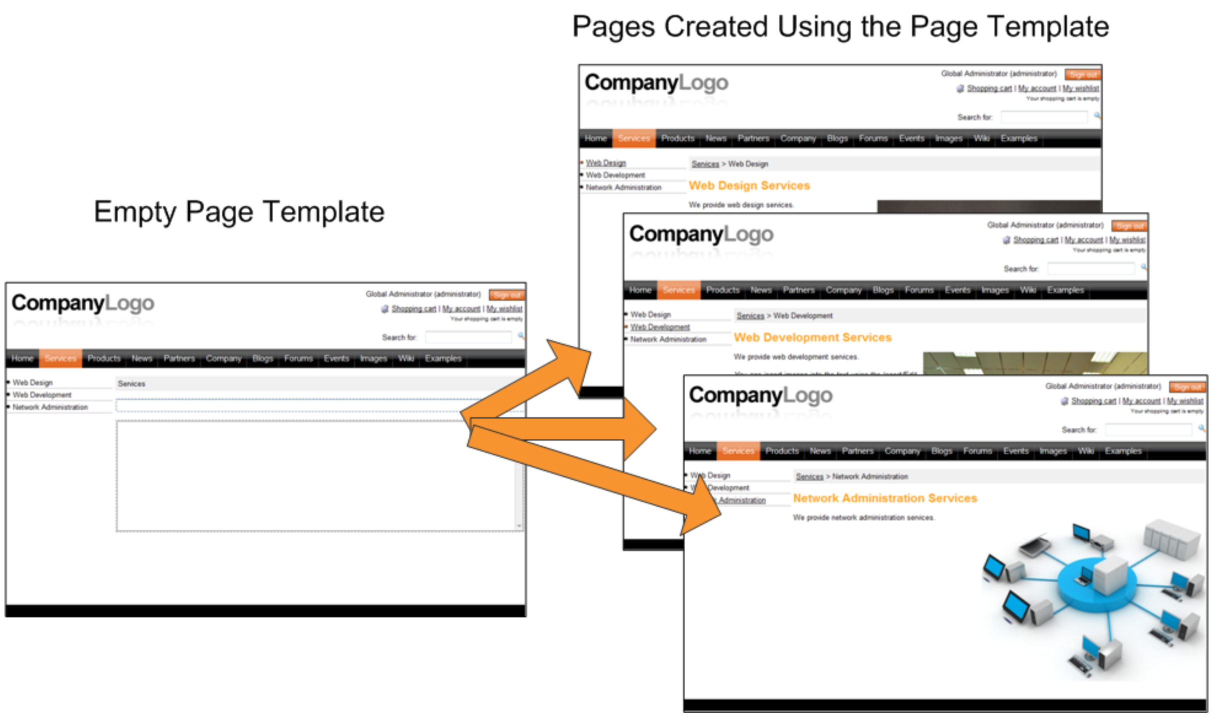
1. The developer analyzes the client requirements.
2. The developer prepares the site structure (site map) and web design.

3. The developer creates **page templates** for every type of the page (home page, solutions, products, news, etc.)
4. The content editor creates new pages - they enter text and images into the page templates defined by the developer.

What is a page template?

The page template is a predefined look of the page that allows content editors to enter the content. A single page template can be re-used for multiple pages with the same structure and design, but with different content.

The templates allow content editors to focus on content editing, without taking care of the page formatting. They also help keep the web design consistent throughout the website. The following figure shows how a single page template can be used for multiple pages:



The details on page template development will be described later.

Part III

Installation and deployment

3 Installation and deployment

3.1 Overview

Kentico CMS installation consists of several steps depending on the installation type:

Installation on the local development machine

- Setup (kenticocms.exe) - installs only the web project files. No changes to the system configuration (registry, IIS or SQL Server) are made. You do not need to run the setup on your production server - it's intended primarily for development machines.
- Web Installer - creates a new website project and optionally configures IIS or uploads the project to the server using FTP. Again, you do not need to run the Web Installer on your production server - you can run it locally and deploy the files over FTP.
- Database setup - runs in the web user interface. It creates a new database on your SQL Server with system tables and basic data.
- New site wizard - runs in the web user interface after you create a new database. It allows you to create a new site managed by Kentico CMS.

Installation on the remote (production) server

On the remote (production) server, it's not necessary to run any executable or register DLL libraries. Unless you have full administrative access to the server, you will typically follow these steps:

- After you install Kentico CMS on your development machine using Setup (KenticoCMS.exe) you need to run Web Installer on your development machine and choose to install Kentico CMS on a remote server. Choose the temporary folder on your local disk.
- Copy the files from the temporary folder to the production server (e.g. over FTP). If the files are not copied directly to the root of the website, you will need to create a virtual directory - see Additional configuration tasks -> Creating a virtual directory for details.
- Open web browser and navigate to the root URL of the copied files on your web server.
- The rest of the installation is the same as on the local machine.

If you encounter any problems during the installation, please see the Troubleshooting installation issues chapter.

Tip: There's no magic behind!

Kentico CMS is a **standard ASP.NET application**. Since it doesn't make any modifications to the system, you can move it to another system as you do with any other ASP.NET project. You can also open the project in Visual Studio and debug it or compile it.

The database is a standard MSSQL database, so you can move it to another server using a common backup/restore procedure. The connection string is stored in the web.config file, in the ConnectionStrings/CMSConnectionString value.

3.2 System requirements

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 XP, 2003, 2008, 2008 R2, Windows Vista Home Premium/Business/Enterprise/Ultimate or Windows 7 (both 32bit and 64bit)
- Microsoft .NET Framework 2.0 or higher
- Microsoft Internet Information Services (IIS) or Visual Studio/Visual Web Developer 2005/2008/2010 built-in web server
- Microsoft SQL Server 2005, 2008 (including free SQL Server Express Edition)

Hosting Requirements

- ASP.NET 2.0 (or higher) and Microsoft SQL Server 2005/2008 support
- Medium-trust or full-trust permissions for the ASP.NET application
- If the server uses medium trust, ASP.NET AJAX 1.0 must be installed on the server.
- It's recommended that your hosting plan comes with 125 MB or more memory and 100+ MB database.

You can use your favorite hosting provider or choose from our hosting partners.

Development Tools

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

Supported Client Browsers for Content Editors

- IE 7.0, IE 8.0
- Firefox 3.5
- Safari 4.0 or Firefox 3.5 on Mac OS

Supported Client Browsers for Site Visitors

- IE 6.0, IE 7.0, IE 8.0
- Firefox 1.0.5+
- Mozilla 1.7.1+
- Netscape 7.1+
- Opera 7.52+
- Safari or Firefox on Mac OS

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

3.3 Installation procedure

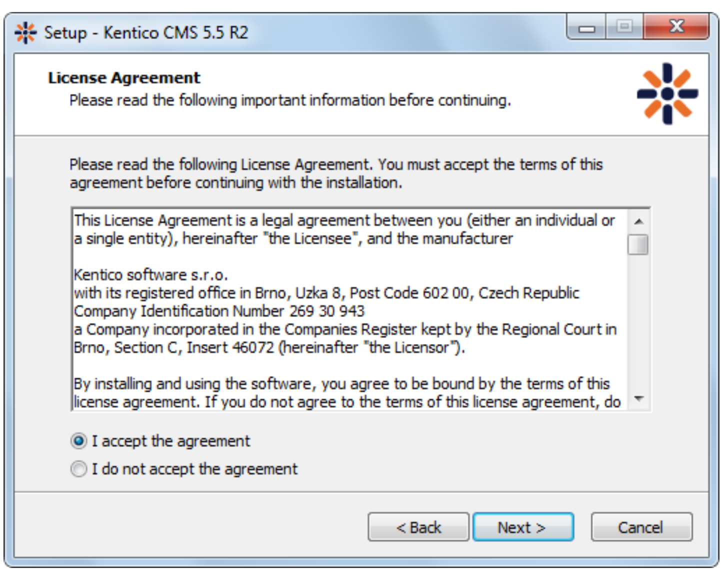
3.3.1 Setup (KenticoCMS.exe)

Installation on shared hosting server

If you're going to run Kentico CMS on a shared hosting server, you do not need to run any EXE file or register any DLL file on the server (you're usually not allowed to do that anyway). Please read Additional configuration tasks -> Installation on shared hosting server to find how to solve this.

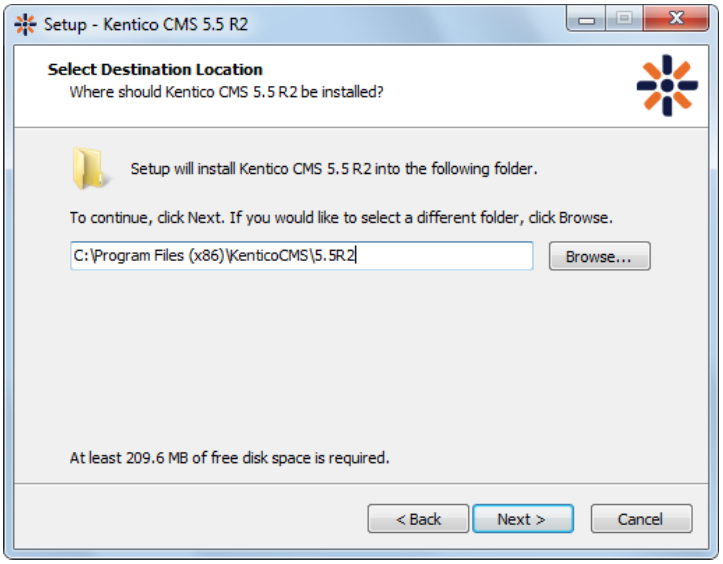
1. Run *KenticoCMS_<version>.exe*. You will see the welcome screen. Click **Next**.

2. Read and accept the license agreement if you want to continue. Click **Next**.



3. Choose the location where Kentico CMS web installer and documentation will be deployed. Click **Next** and then **Install**.

Please note: this is not the folder where your website will be placed, it's only a place for Kentico CMS program files and help files.



4. After the installation is finished, enable the **Launch Kentico CMS Web Installer** option and click **Finish**. Continue to Kentico CMS Web Installer.

3.3.2 Web installer

3.3.2.1 Overview

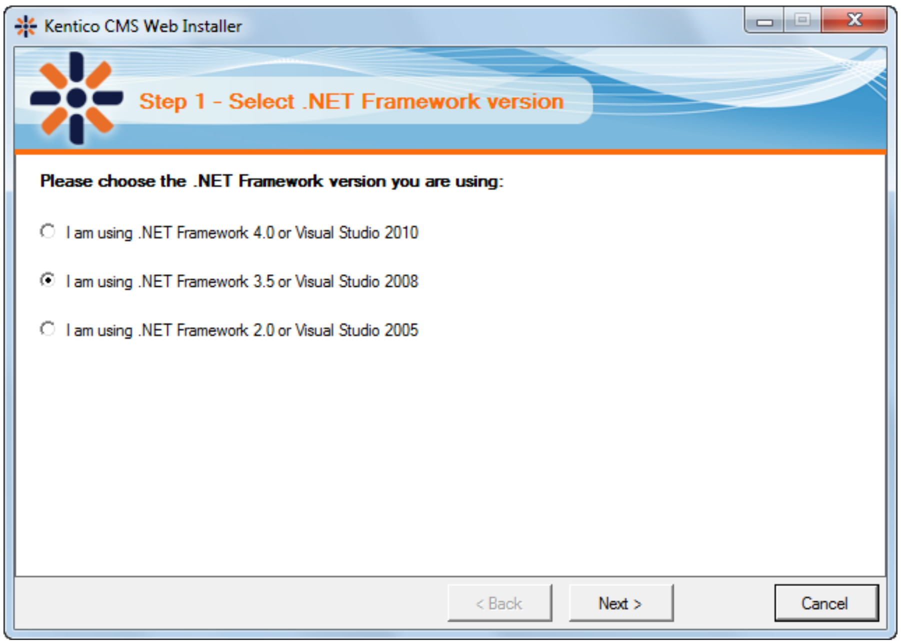
Kentico CMS Web Installer allows you to create a new project and (optionally) configure Microsoft IIS web server.

Step 1 - Select .NET Framework version

First, you need to choose whether you use:

- **.NET Framework 4.0 and Visual Studio 2010**
- **.NET Framework 3.5 and Visual Studio 2008**
- **.NET Framework 2.0 and Visual Studio 2005**

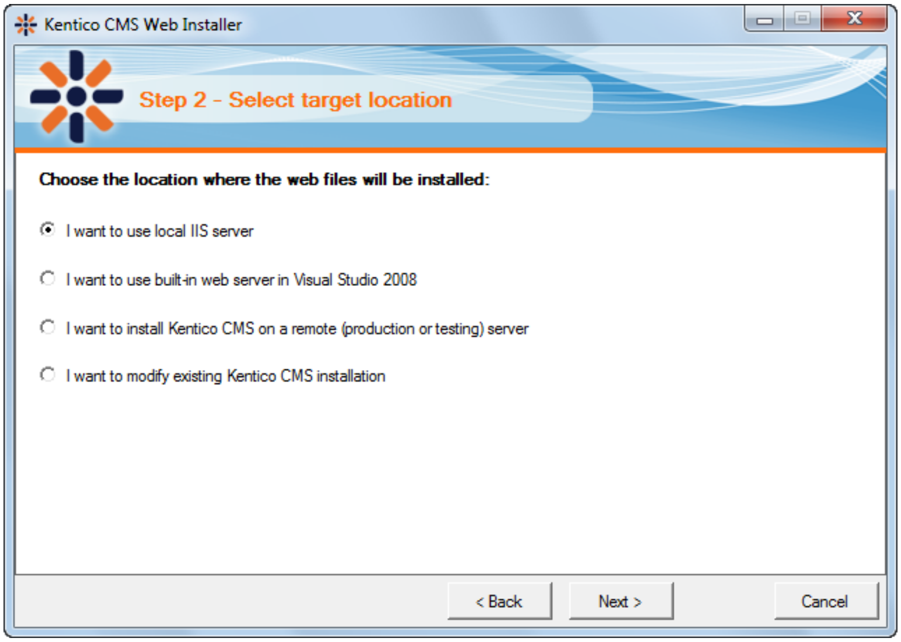
Depending on your choice, the installer will use appropriate web.config and webproject.sln files. The binaries and code are the same for all options; they are compiled for .NET 2.0 and can be used with .NET 3.5 or .NET 4.0 as well.



Step 2 - Choosing the target location

Choose one of the options. Click the link to view instructions for the selected option.

- I want to use local IIS server - you must have local IIS server installed, running and configured for the version of ASP.NET which you chose in the previous step
- I want to use built-in web server in Visual Studio - you must have Visual Studio or Visual Web Developer Express Edition installed on your local machine
- I want to install Kentico CMS on a remote (production or testing) server - this option only copies the project files to a temporary folder on your disk and you need to copy the files to your production server manually (e.g. over FTP)
- I want to modify existing Kentico CMS installation - this option modifies (adds or removes components) an existing installation on the local machine



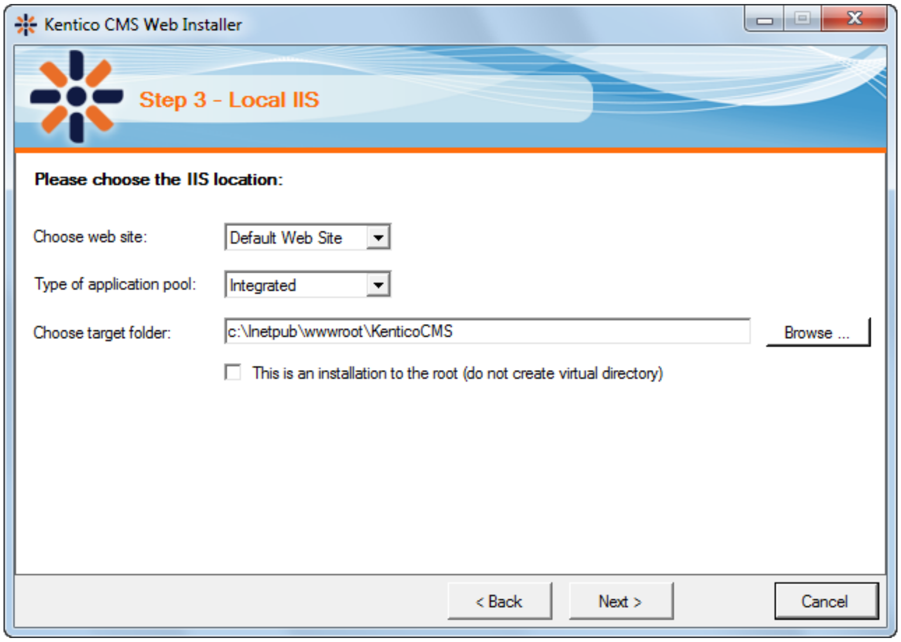
3.3.2.2 Local IIS server

If you run local IIS server on your machine, choose **I want to use local IIS server** in Step 2 of the Web installer and click **Next**.

1. In Step 3, you need to specify the following information:

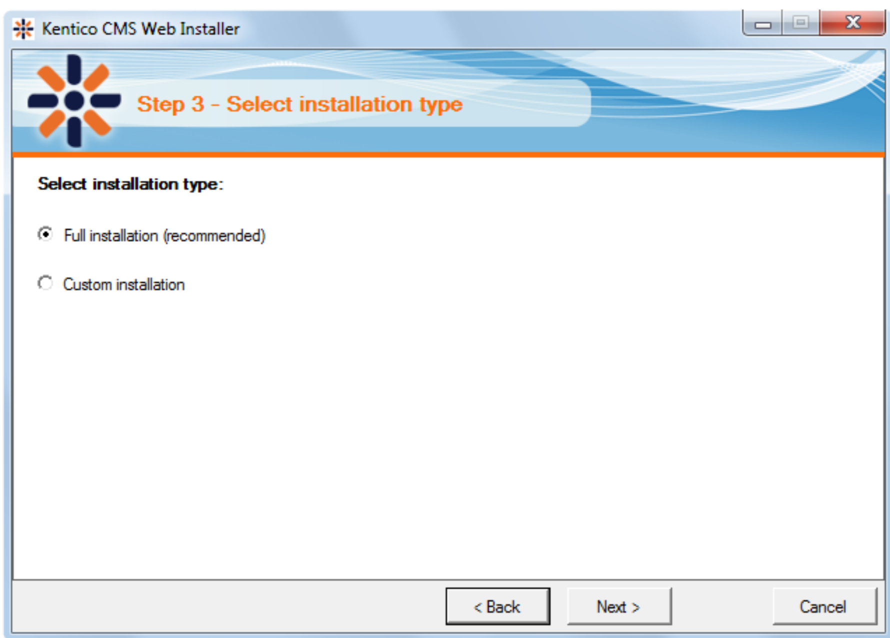
- **Choose website**: choose one of the websites configured on your IIS; please make sure that the website you choose is running
- **Type of application pool**: type of application pool which will be used for the website (*Integrated* by default, read here for more information)
- **Choose target folder**: choose the disk folder where web project files will be placed

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

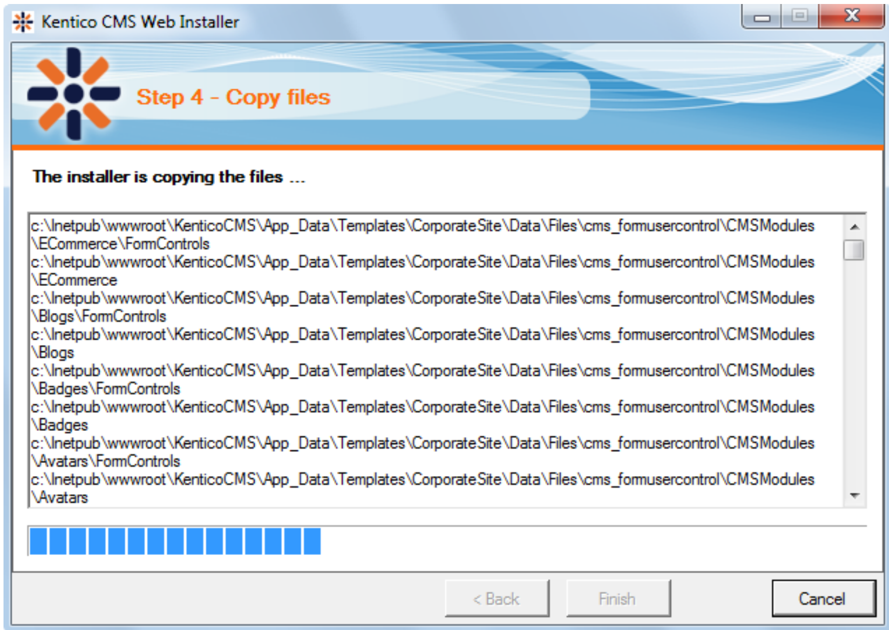


2. Now you need to decide between the following options:

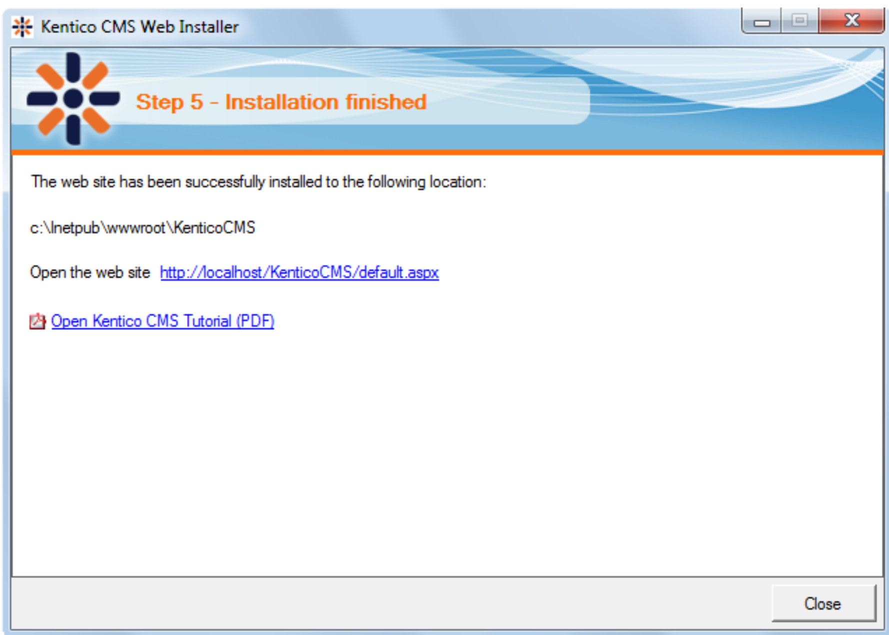
- **Full installation** - performs full installation using all optional components
- **Custom installation** - lets you choose which components will be installed



3. Now the installer copies the project files to the specified folder:



4. At the end, you're displayed with URL of the new Kentico CMS installation. Please put down this URL for future use.



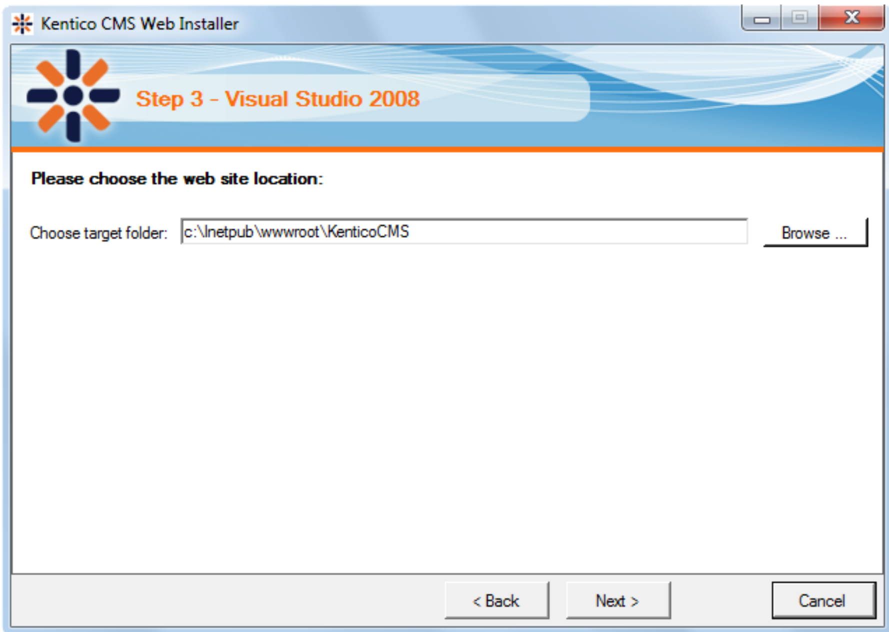
5. Now you can continue to Database setup wizard.

3.3.2.3 Built-in web server in Visual Studio

If you want to use the build-in server which is included in Visual Studio, choose **I want to use built-in web server in Visual Studio** in Step 2 of the Web installer and click **Next**.

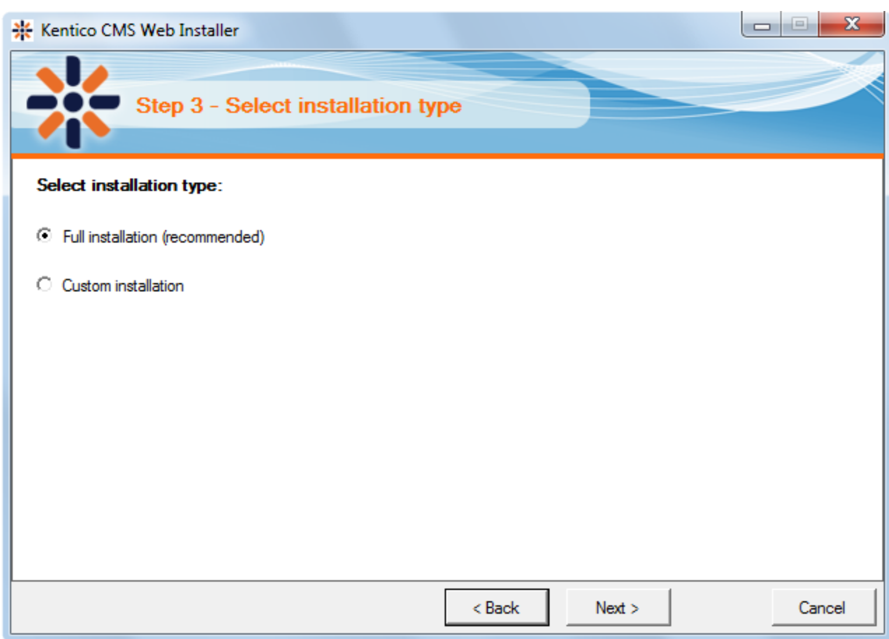
1. You will need to enter the target location of the files on your disk in Step 3 and proceed by clicking

Next.

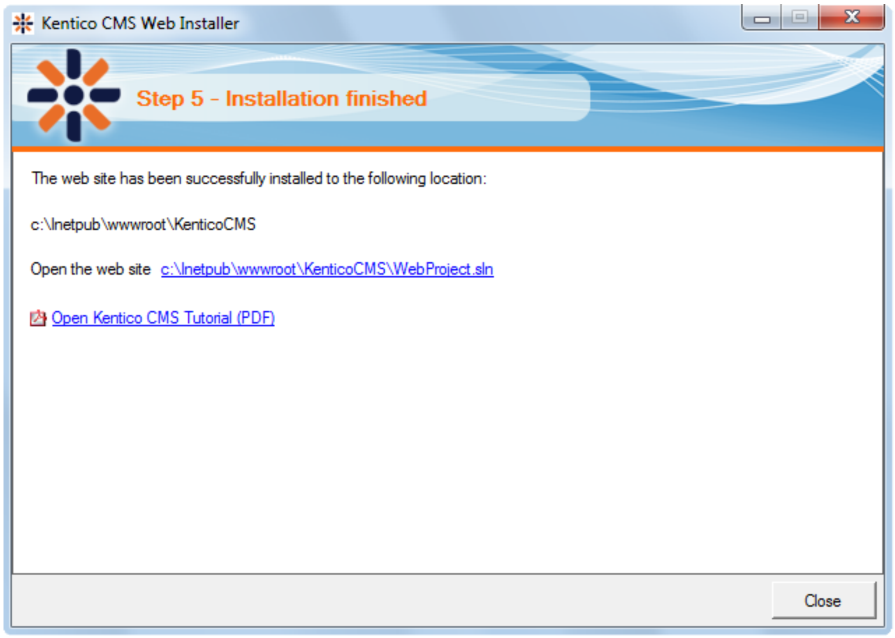


2. Now you need to decide between the following options:

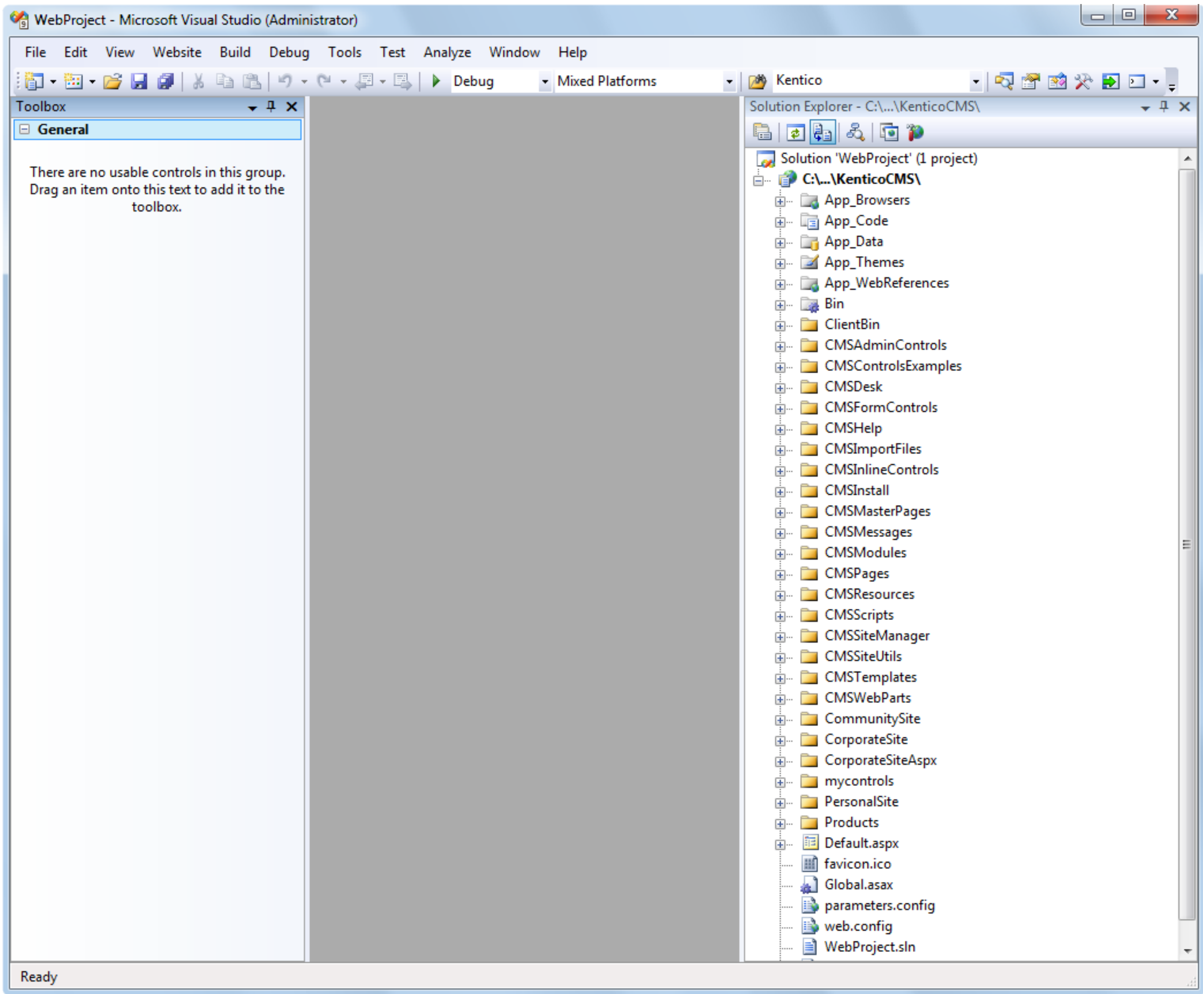
- **Full installation -** performs full installation using all optional components
- **Custom installation -** lets you choose which components will be installed



3. After the files are copied to the folder, you can open the solution in Visual Studio by clicking the link:



4. The project will open in Visual Studio:



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

When you cannot open the website in Visual Studio

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

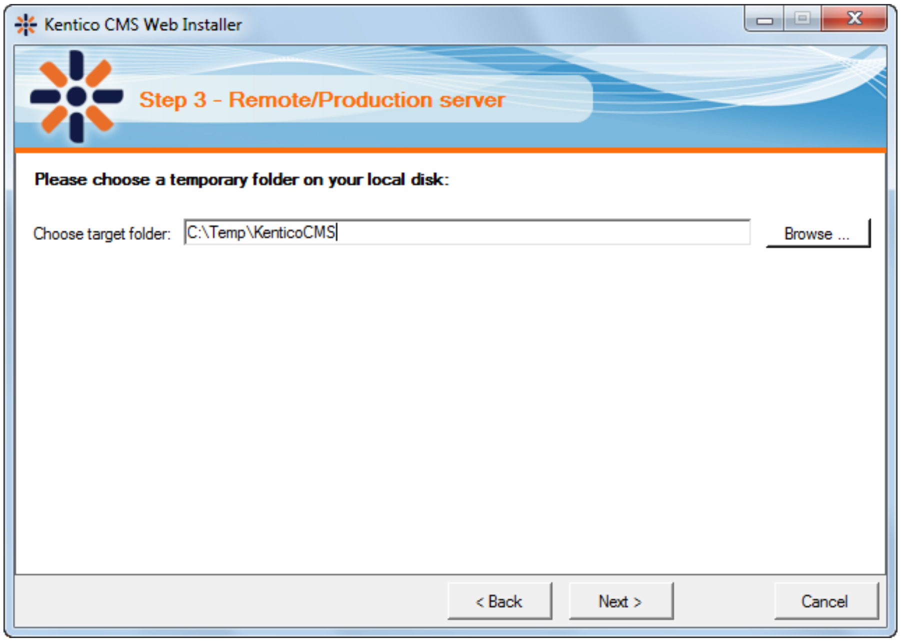
6. Now you can continue to Database setup wizard.

3.3.2.4 Remote server

If you need to install Kentico CMS on a remote web server where you cannot run the setup directly (e.g. a shared hosting server), you need to choose the **I want to install Kentico CMS on a remote (production or testing) server** option in Step 2 of the Web installer.

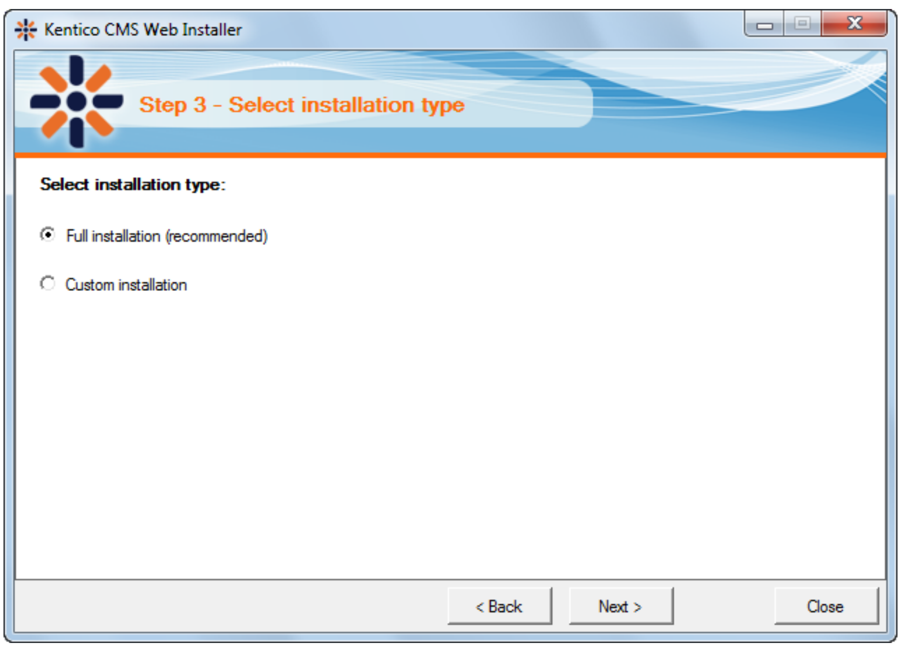
1. In Step 3, you will need to specify a temporary folder on your local disk where the web project will be

created. Proceed through the rest of the wizard.

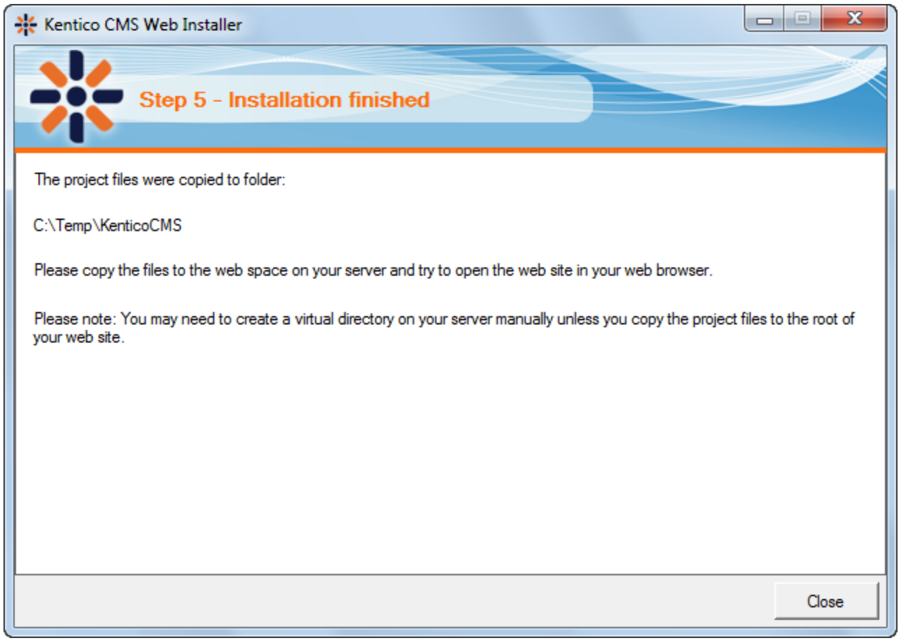


2. Still in Step 3, you need to decide between the following two options:

- **Full installation** - performs full installation using all optional components
- **Custom installation -** lets you choose which components will be installed



3. When you reach Step 5, the web project is created on your disk and a confirmation message is displayed:



4. You need to copy the website to your server (e.g. over FTP). If your web project isn't placed in the root of the remote website, you may need to create a virtual directory as described in Creating a virtual directory.

Application pool settings

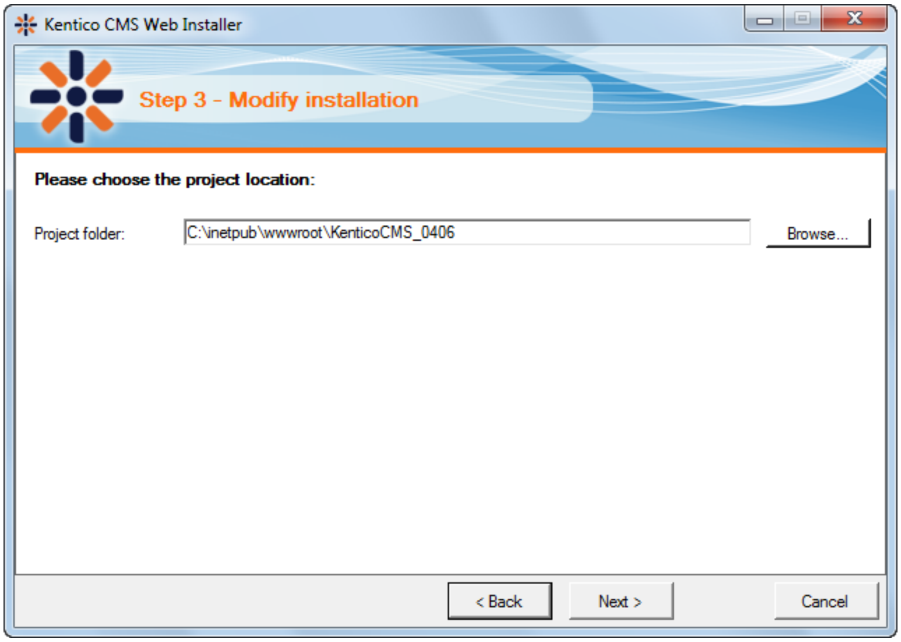
By default, the **web.config** file of the created web project is configured to use an application pool with *Classic* managed pipeline mode. If you know that the remote server uses *Integrated* pipeline mode, replace the default web.config file as described in Troubleshooting installation issues -> Internal server error when using Integrated pipeline mode.

5. Now you can continue to Database setup wizard.

3.3.2.5 Existing installation

The **I want to modify existing Kentico CMS installation** option in Step 2 of the Web installer enables you to add or remove components in an existing Kentico CMS web project.

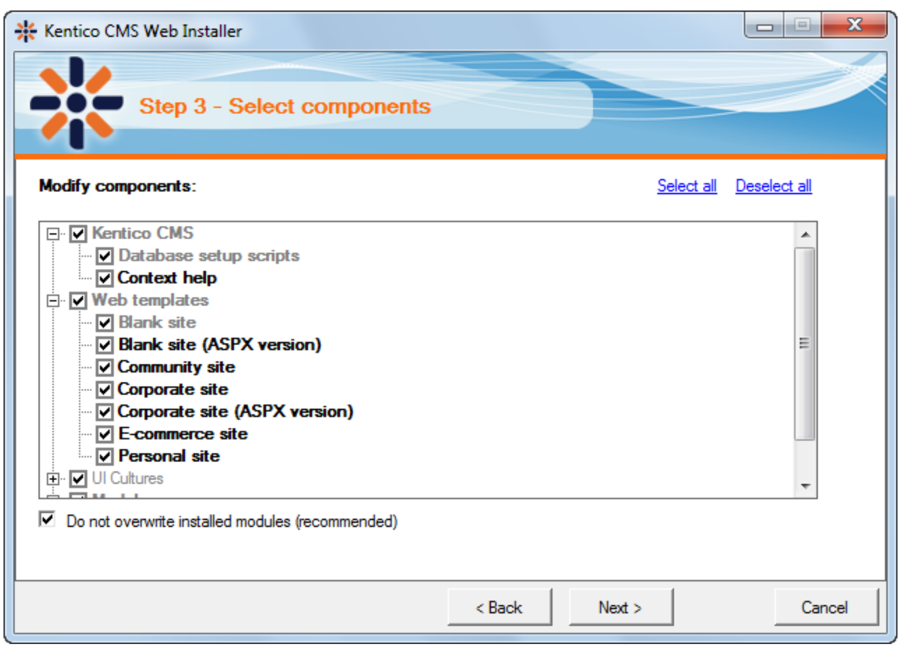
1. In Step 3, you need to specify the root folder of the project that you want to modify. Enter the path to the project folder and click **Next**.



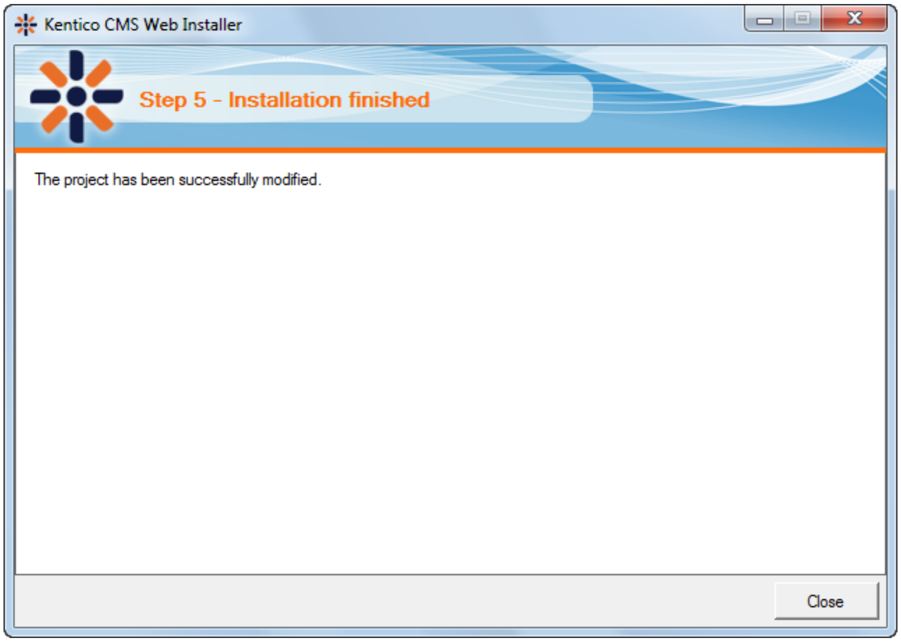
2. The following step displays the component tree. Gray components are a native part of the installation and can not be removed. Black components are optional. The check-boxes indicate which components are currently installed and you can modify the installation by selecting or unselecting them.

If you enable the **Do not overwrite installed modules (recommended)**, only unselected components will be removed and additionally selected components will be added. If the option is disabled, all enabled components (including *web.config*) will be overwritten by the installation, which means that your modifications to these components may get lost!

Click **Next**.



3. When you reach the following screen, it indicates that your project has been successfully modified.



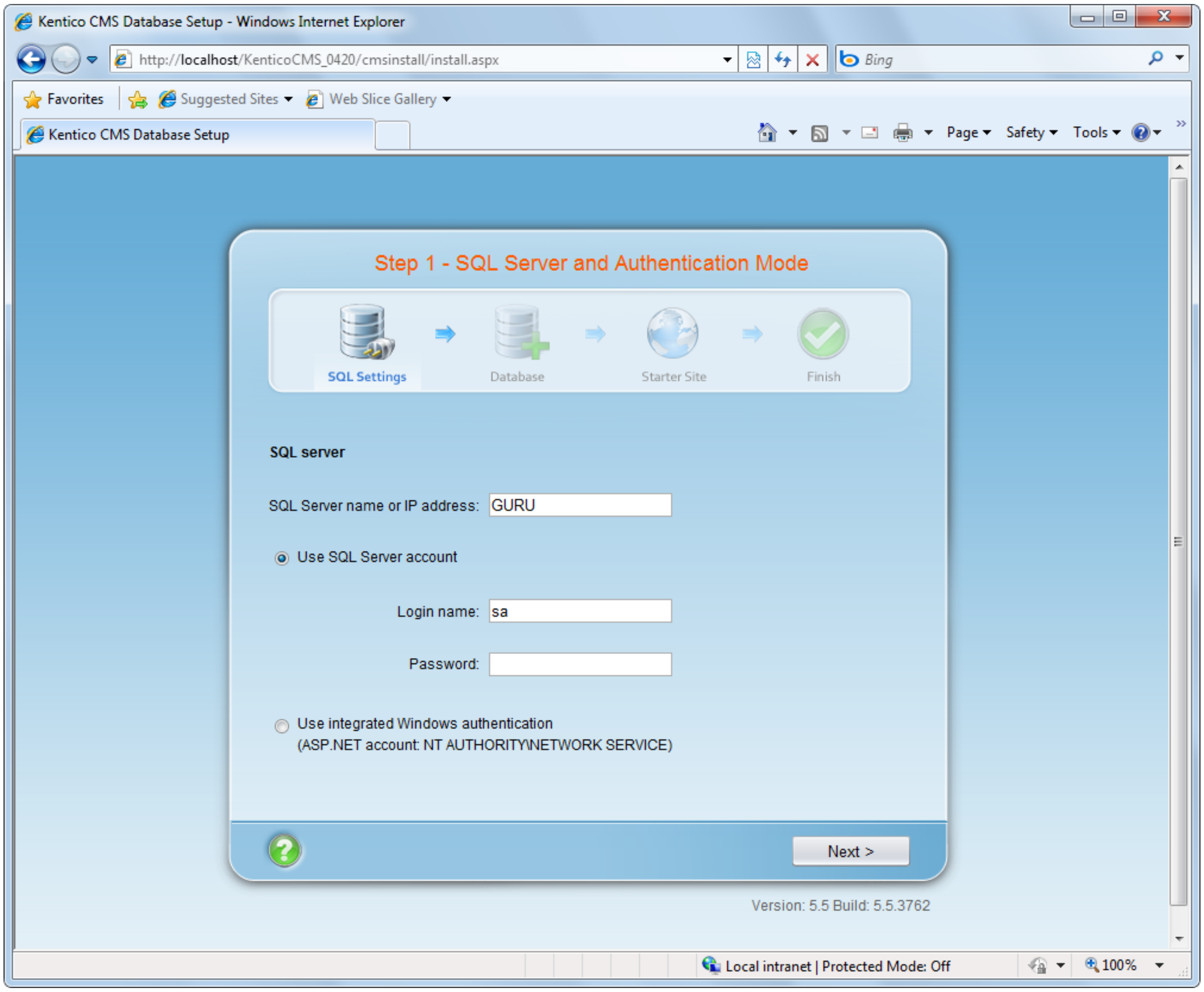
3.3.3 Database setup

After you finish the Web Installer successfully, you open a web browser with the Database Setup. It will create database tables for Kentico CMS in a specified SQL Server database.

1. In Step 1, you need to specify the SQL Server and authentication mode used to access the server:

- **SQL Server name or IP address -** enter the name of the server. You will typically use one of these:
  - the name of the server (such as DBSERVER1) or
  - the IP address of the server (such as 192.168.1.105) or (local)
  - *<SERVERNAME>\sqlexpress* (if you're using Microsoft SQL Server 2005 Express Edition)
- **Use SQL Server account -** use this option if your server is configured for **mixed mode** authentication with SQL logins
- **Use integrated Windows authentication (ASP.NET account) -** use this option if your server is configured for Windows integrated authentication. In this case, you need to use SQL Server 2005/2008 Management Studio to create a new login for user account under which you currently run the web application (ASPNET for Windows XP and Network Service for 2003 - the actual ASP.NET account name is displayed on the screen).

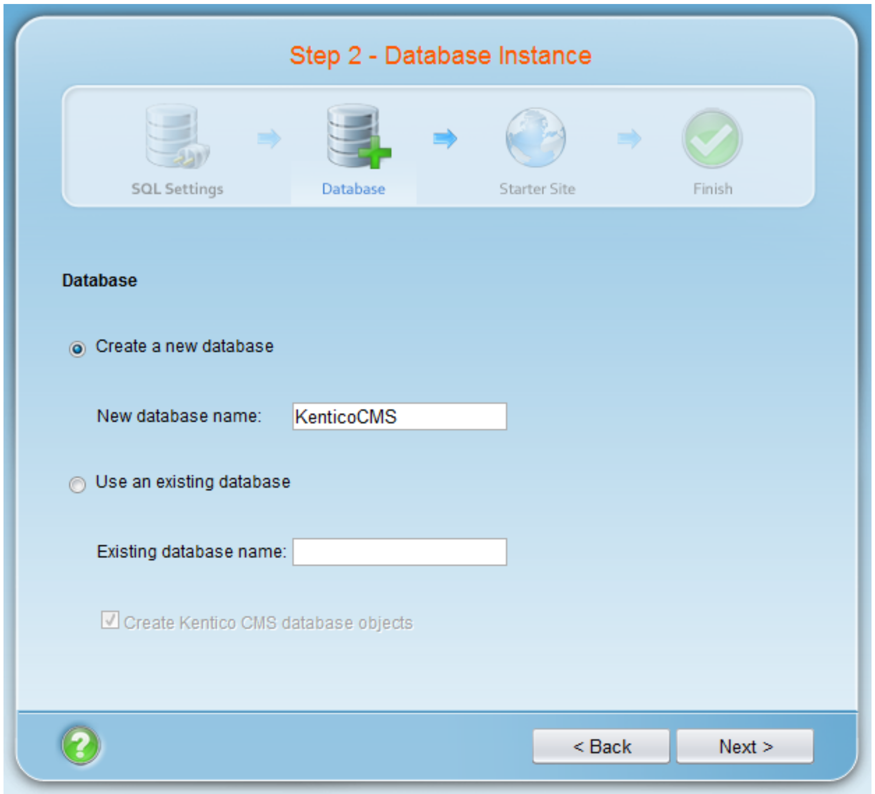
Click the **Next** button.



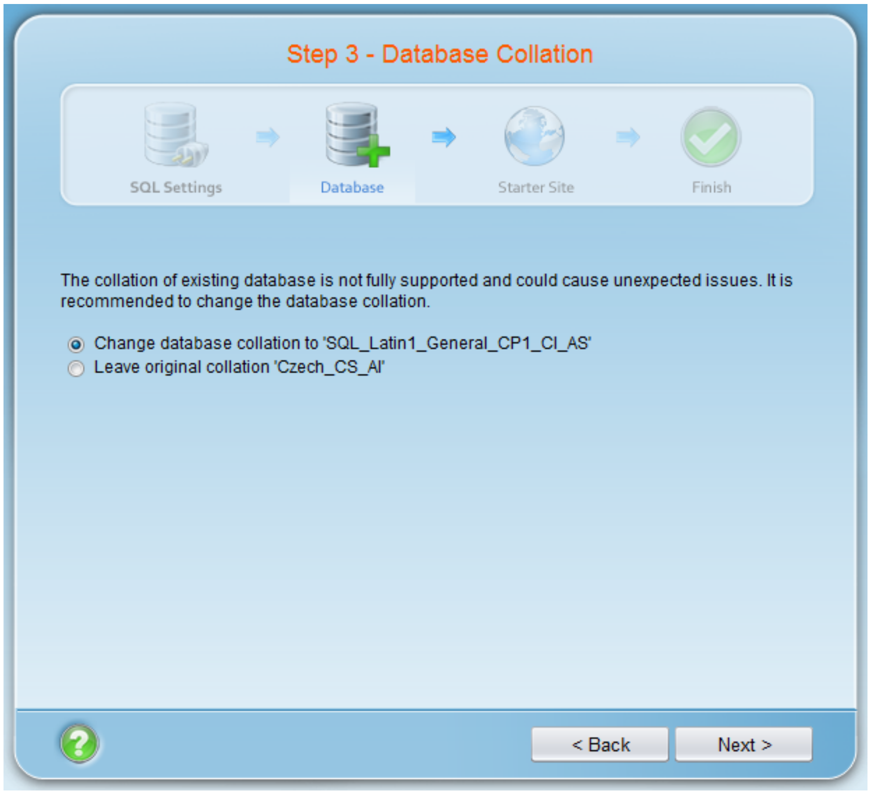
2. Now you can decide if you want to use an **existing database** or **create a new database**. In both cases, you need to enter the name of the database into the appropriate field.

In case you are using an existing database, you can choose if you want to **Create Kentico CMS database objects**. If the existing database already contains Kentico CMS objects (tables, stored procedures, views), you need to uncheck the box. If the database does not contain these objects (typically when you are installing into an empty database), you need to leave the option enabled.

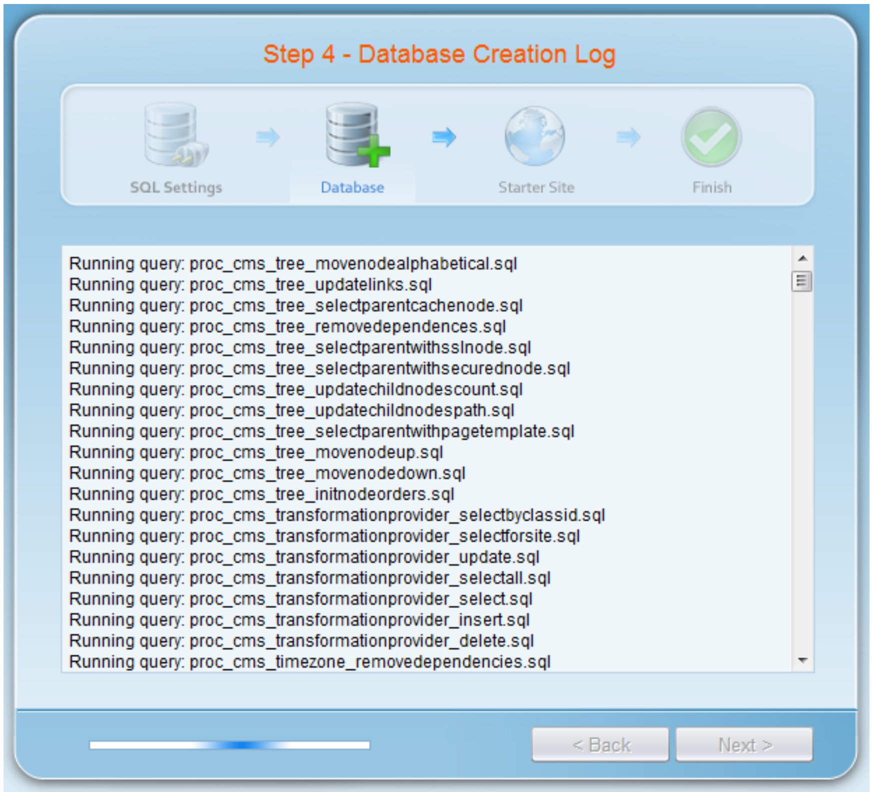
Click **Next.**



3. When using an existing database, you may also come across the following dialog. It is displayed in case that the database collation is different than *SQL_Latin1_General_CP1_CI_AS*. The dialog lets you choose if you want to change the collation or leave the original one. For correct functionality, it is highly recommended to change it to the recommended value.



4. A log will be displayed, showing the progress of database creation. After it finishes, a message *Database has been successfully created.* appears at the top of the log and you are moved forward to the next step.



5. If you run Kentico CMS on a domain other than *localhost* or *127.0.0.1*, you will be asked for inserting a license key, since the trial version works only with *http://localhost* and *http://127.0.0.1*. The same dialog is displayed if your trial period has expired.

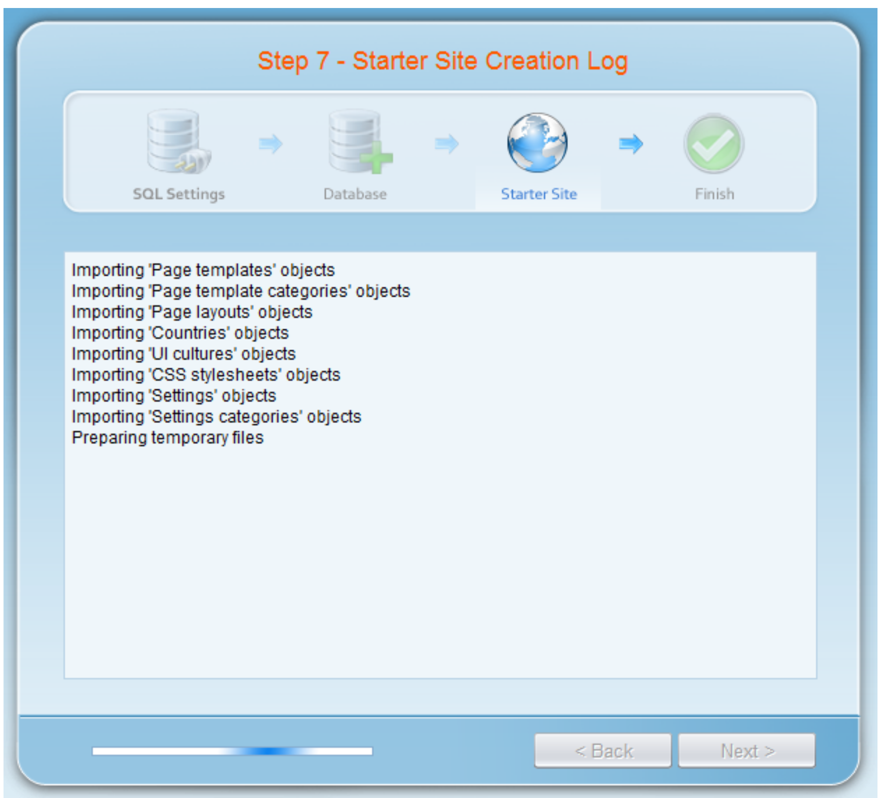
Enter a valid license key and click the **Next** button. Alternatively, you can skip this dialog and go to **Site Manager -> Sites -> New site wizard** by clicking **Skip this dialog** at the bottom left part of the dialog.

6. On the next screen, you will be offered with the following options:

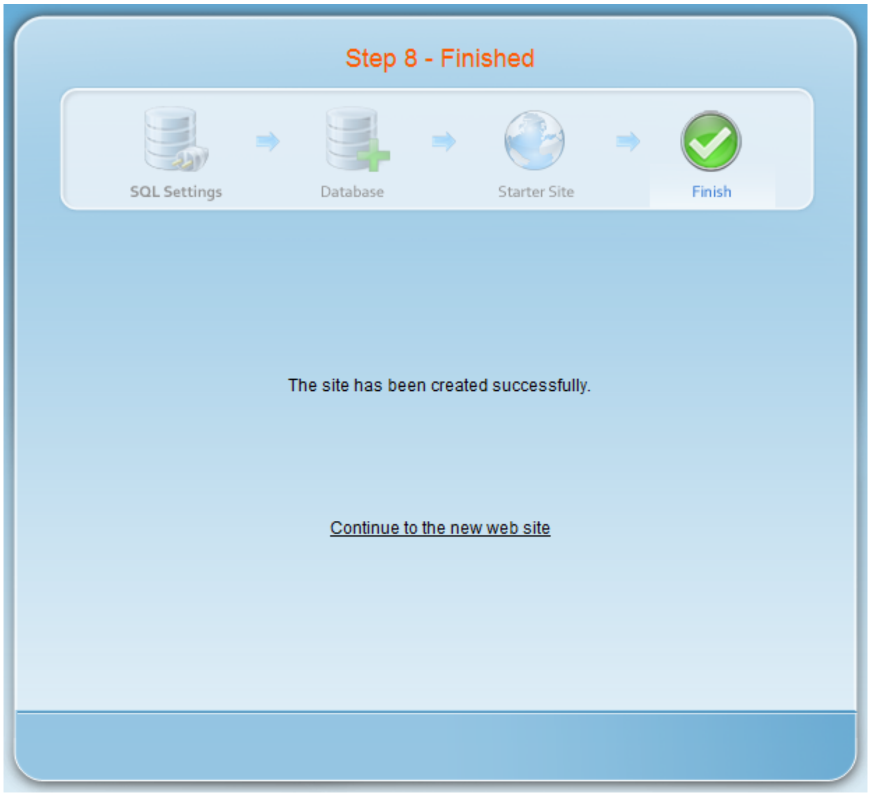
- **Choose a starter site such as:**
  - **Corporate Site** - this option installs the sample corporate site - it is recommended for most users, especially for evaluators
  - **Corporate Site ASPX** - this option is recommended only for experienced ASP.NET developers who want to use ASPX page templates instead of portal templates
  - **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
  - and others
- **Continue to the New site wizard** - this option is recommended if you're starting a new site from scratch; learn more in the New site wizard chapter
- **Import an existing Kentico CMS website** - use this option is you already created a website with Kentico CMS and need to import it into a new installation (e.g. on the production server)

If you're new to Kentico CMS, it's highly recommended that you start with sample Corporate Site (portal engine). If you decide to run the New site wizard, you can find more details in the following chapter. Select an option and click the **Next** button.

7. A log will be displayed in the next step, showing you progress of website creation.



8. Once the website is created, you will be displayed with a confirmation and link to your live website.



Default user name and password

The default user name is **administrator**, the default password is blank.

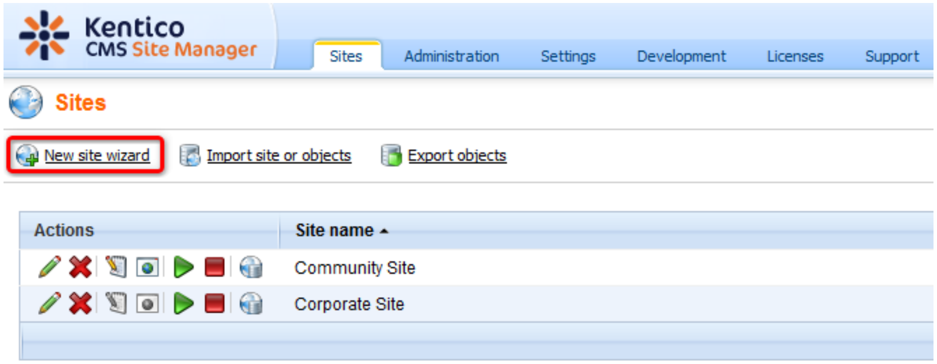
It's highly recommended that you change the password after you finish the installation.

The default URL of your site is *http://localhost/KenticoCMS*.
The default URL of CMS Desk is *http://localhost/KenticoCMS/CMSDesk*.
The default URL of Site Manager is *http://localhost/KenticoCMS/CMSSiteManager*.

3.3.4 New site wizard

3.3.4.1 Overview

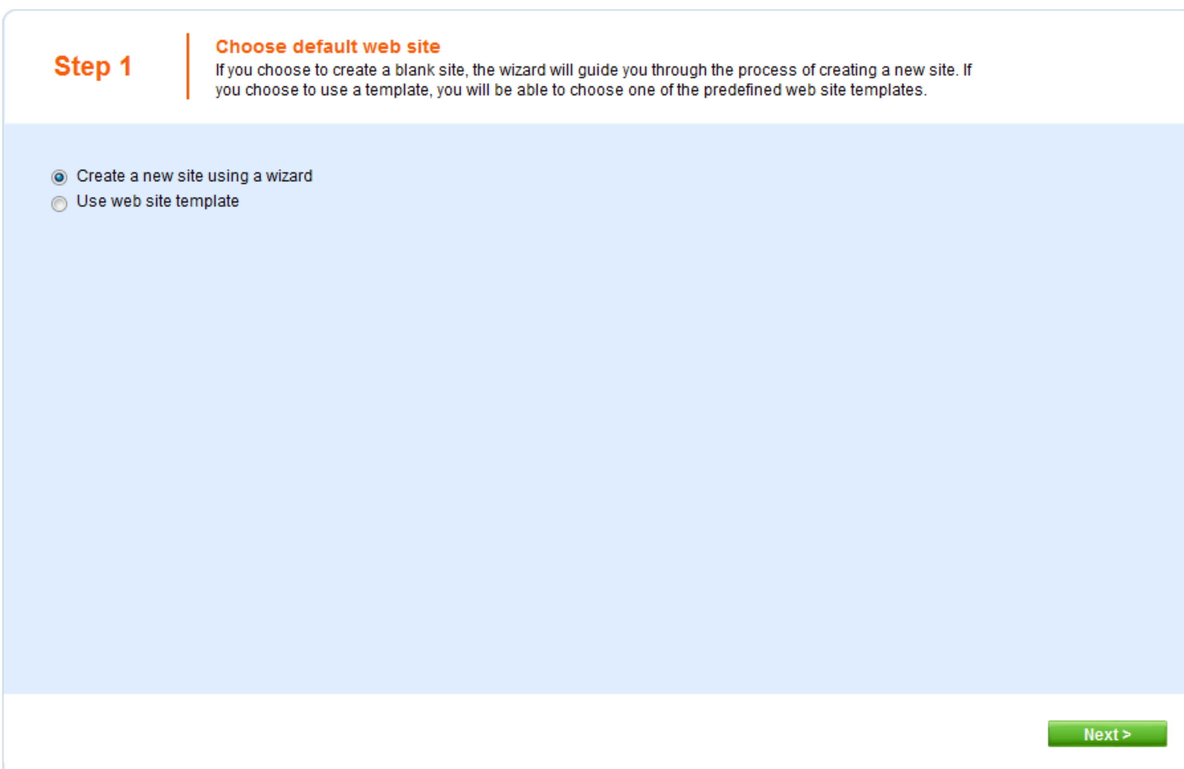
The New site wizard guides you through the process of adding a new website to the system. It is accessible from the *Starter Site* step of the Database setup. You can also launch the wizard from **Site Manager -> Sites**.



In the first step of the wizard, you can select if you want to create the new site using a wizard or use a website template.

- Create a new site using a wizard - creates a new blank site and allows you to configure the basic structure of the site.
- Use website template - creates a new site based on a template chosen in the next step.

Select one of the options and click the **Next** button. Click one of the links above to learn about the following steps of the wizard for the chosen option.



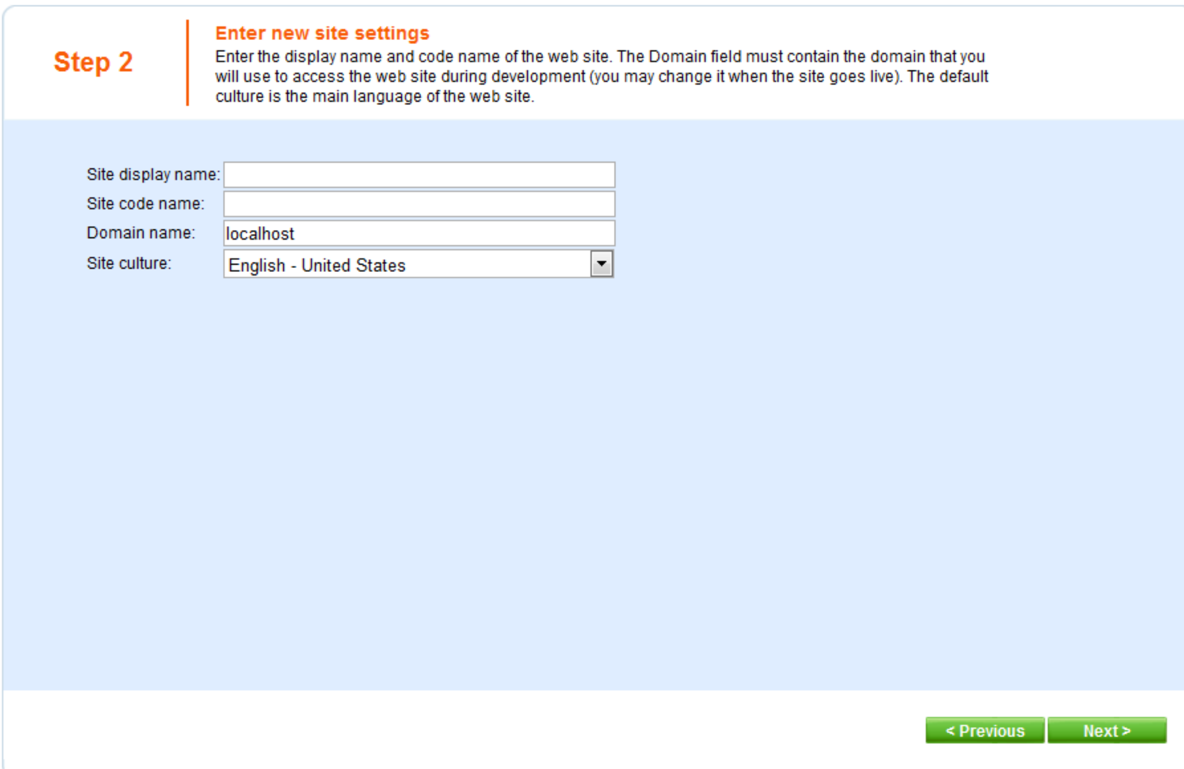
3.3.4.2 New site

This topic describes adding of a new website to the system using the **New site wizard** when the **Create a new site using a wizard** option is chosen in the first step of the wizard.

1. In the second step, you need to enter the following basic site properties:

- **Site display name** - name of the new site displayed in the administration interface.
- **Site code name -** name of the new site used in website code.
- **Domain name** - domain name on which the new site will be running.
- **Site culture** - default culture of the site.

Click **Next** to continue.



2. In Step 3, you can select which objects will be imported along with your new site. You can make this selection by choosing one of the categories displayed in the tree view on the left side of the screen. By selecting a category, a set of check boxes appears in the right part of the screen, letting you select the exported objects.

If you select the root of the tree, you will be offered with the following options:

Global selection

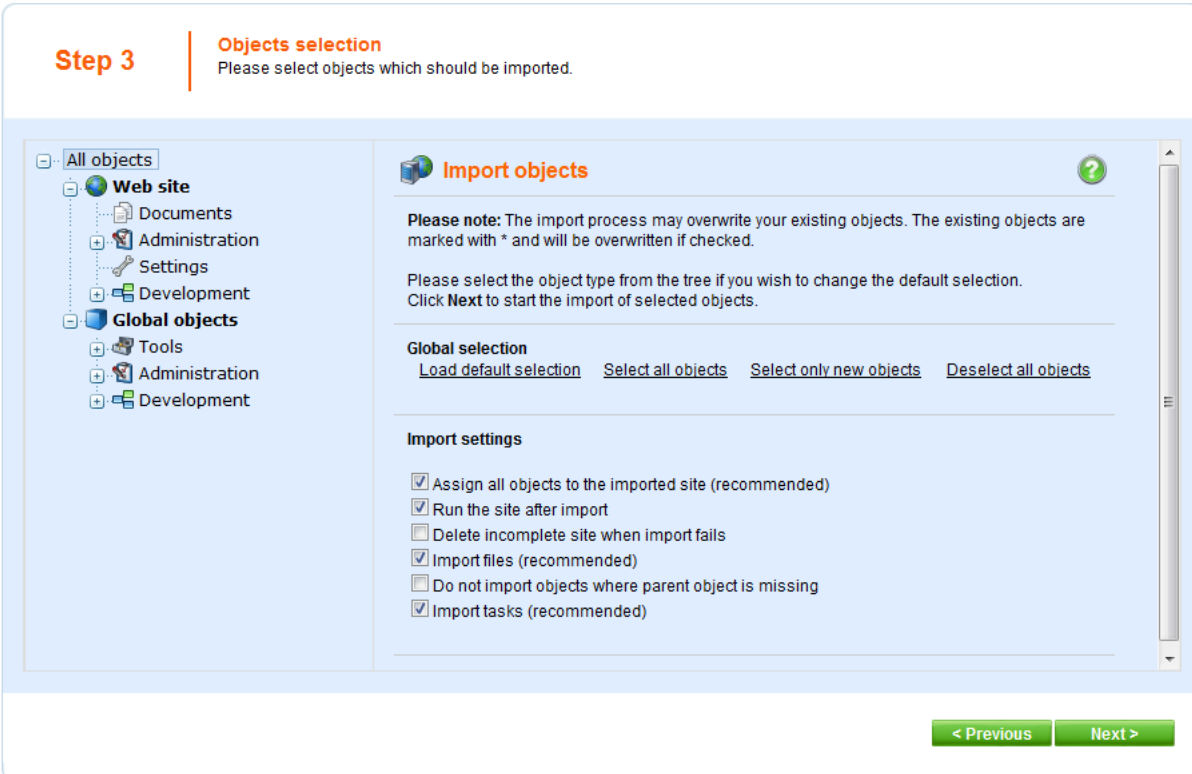
- **Load default selection -** if clicked, object pre-selection will be done based on choice in Step 1.
- **Select all objects -** if clicked, all objects will be preselected.
- **Select only new objects -** if clicked, only objects not existing in the database will be preselected.
- **Deselect all objects** - if clicked, all objects will be de-selected.

Import settings

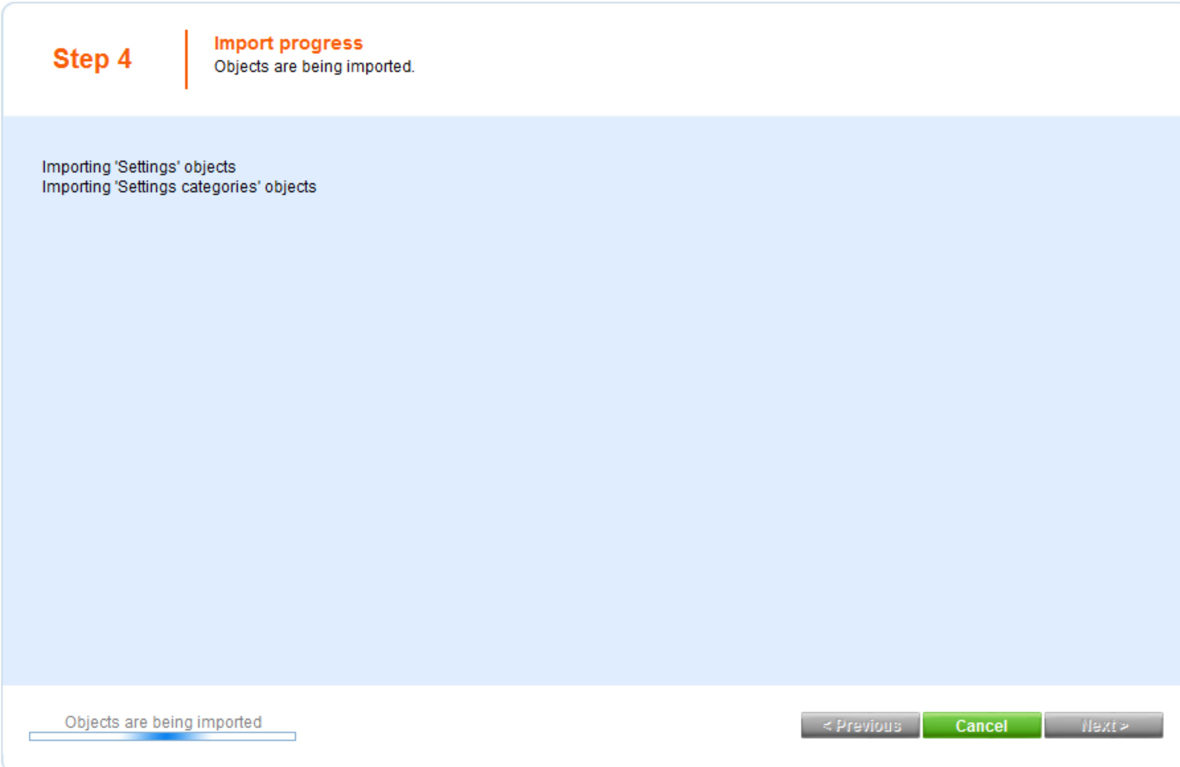
- **Assign all objects to the imported site (recommended) -** if checked, all imported site related objects will be assigned to the imported site.

- **Run the site after import** - if checked, the updated site will be run after the import is finished.
- **Delete incomplete site when import fails** - if checked, incomplete site will be deleted when import fails.
- **Import files (recommended)** - if checked, files will be imported.
- **Do not import objects where parent object is missing** - if checked, objects that are in the package but whose parent object is not present in the target instance will not be imported
- **Import tasks (recommended)** - if checked, delete tasks (incremental deployment) included in the package will be imported

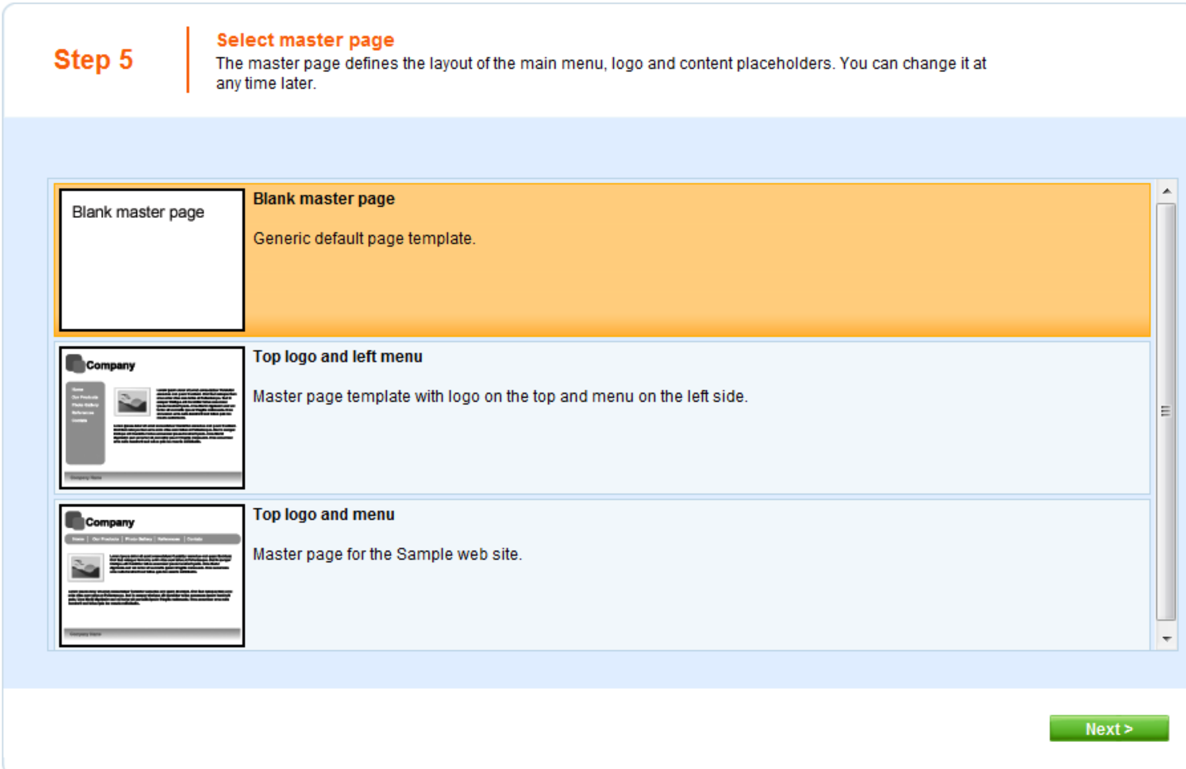
Click **Next** to continue.



3. A log will be displayed, showing you the progress of site import. When the export process finishes, click the **Next** button.

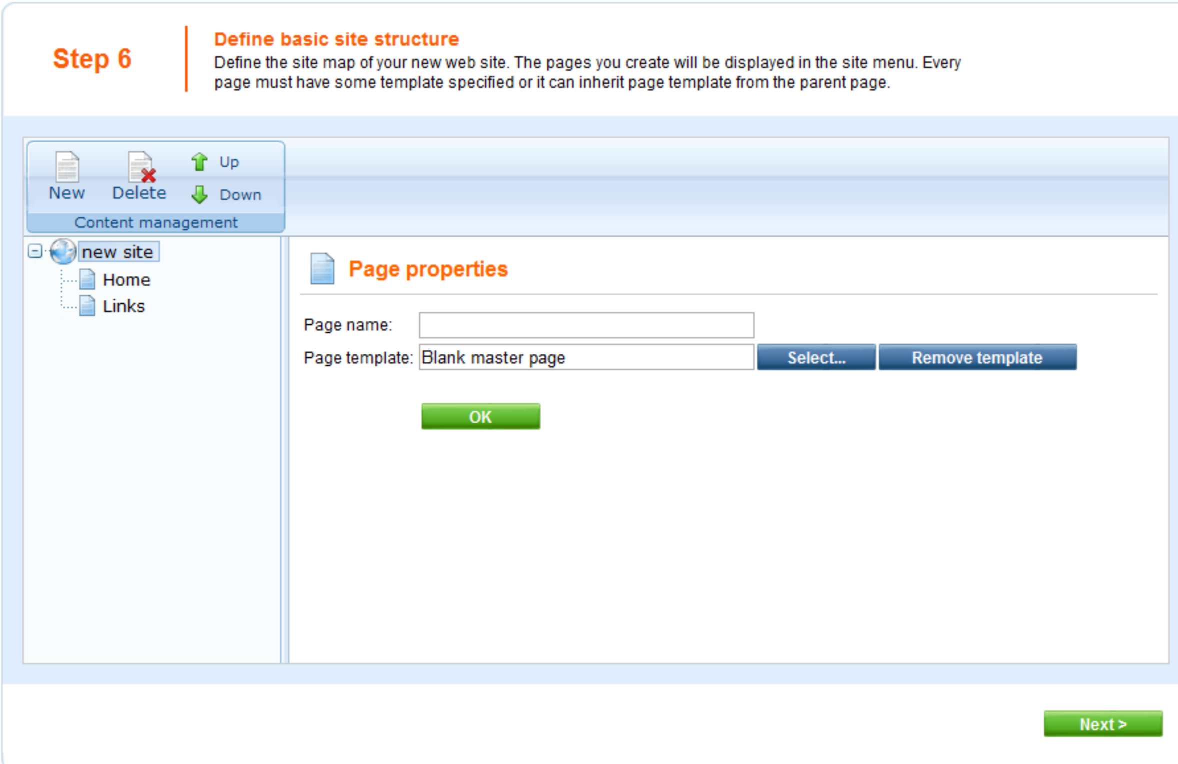


4. In Step 5, you can select the master page layout. This defines the basic visual structure of the website. These settings can be altered any time later, no matter which layout you have selected. Select one of the layouts and click **Next**.

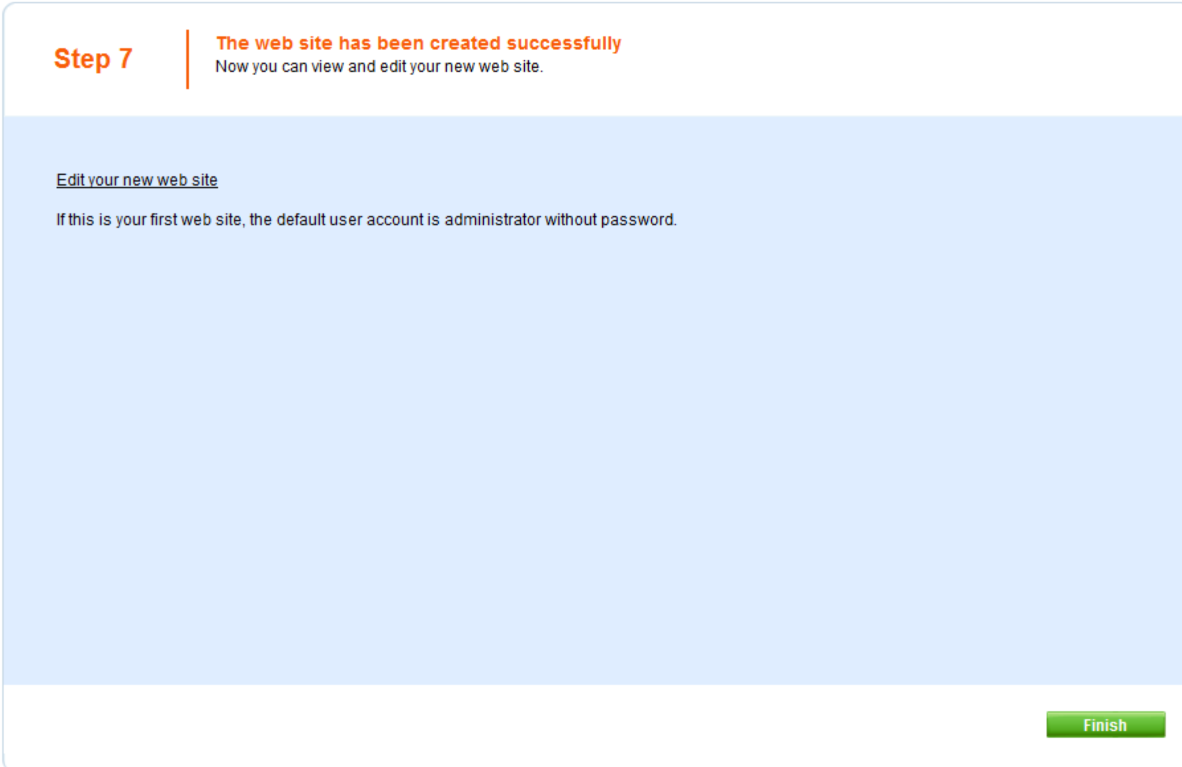


5. Site map of you new website can be defined in Step 6. Select the node of the tree under which you want to place a new page and click **New**. Enter the name of the new page and select a page template which will be used on the page. Alternatively, you can inherit the page template from the parent page by clicking the **Inherit template** button. Click **OK**.

The newly created page will appear in the tree view. Repeat this procedure until you have defined the desired site structure, then click **Next**. The defined site structure can be modified later.



6. If you have reached Step 7, you have successfully created the new website. Click the **Edit your new website** link to switch to CMSDesk and start editing the site immediately. Alternatively, click the **Finish** button to get redirected to **Site manager -> Sites**.



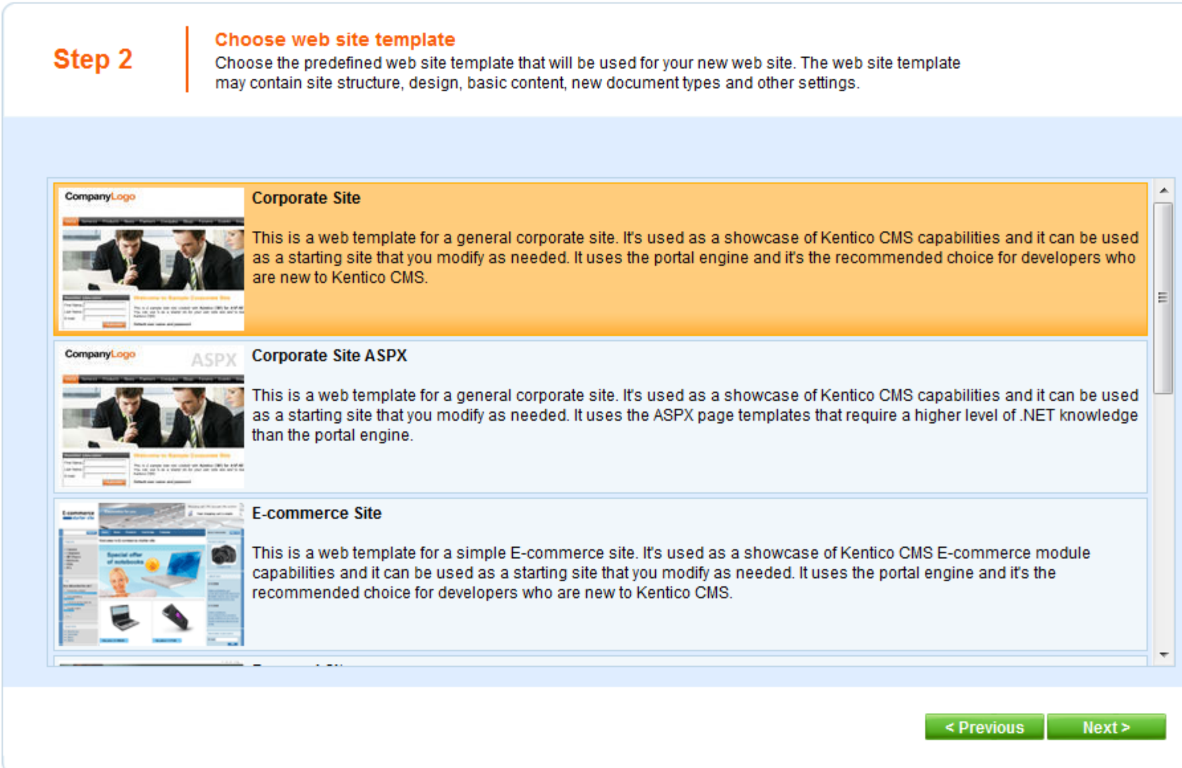
3.3.4.3 Website template

This topic describes creation of a new website using the **New site wizard** when the **Use website template** is chosen in the first step of the wizard.

1. In Step 2, you can choose from a number of website templates:

- **Corporate site** is a typical web presentation of a company.
- **E-commerce site** is a typical e-shop showing the possibilities of the E-commerce module.
- **Community site** is a sample community website demonstrating social networking capabilities of Kentico CMS.
- **Blank site** is a blank template used for creating websites from scratch.
- and others.

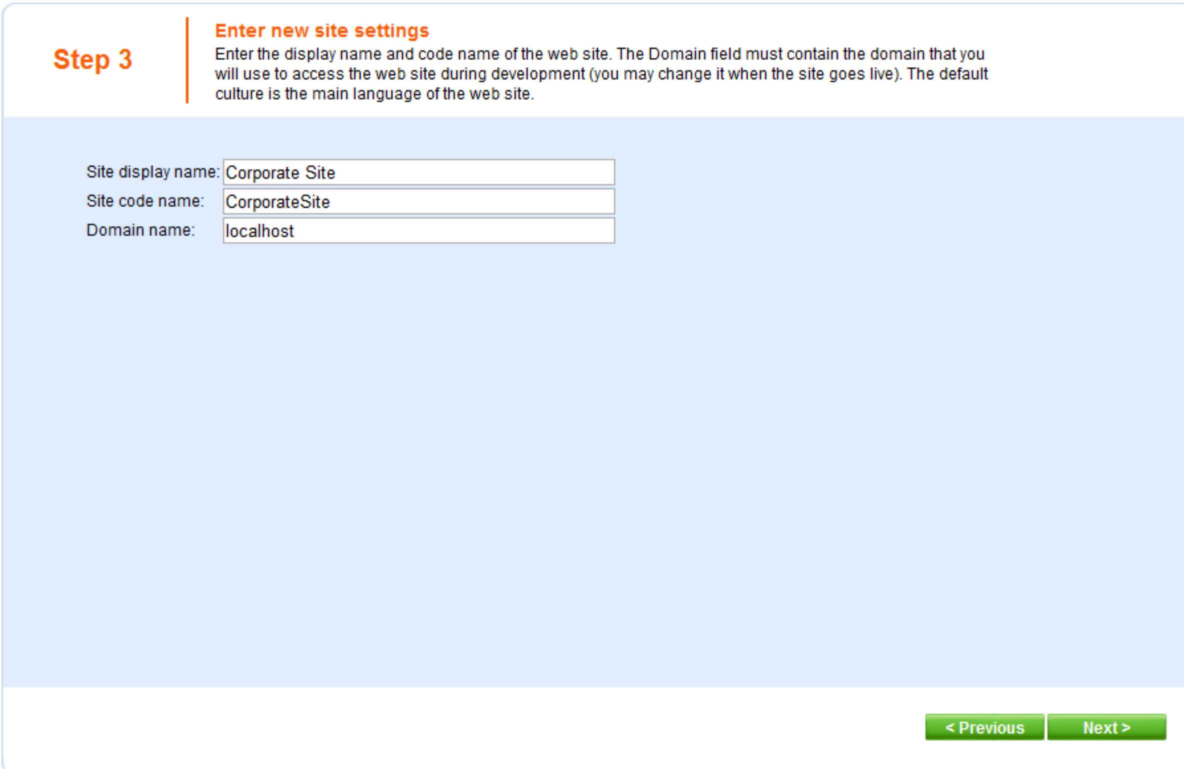
Some of the templates are available in two versions, one using the portal engine and the other using ASPX page templates. Choose one of the listed website templates and click the **Next** button.



2. In this step, enter the following basic site properties:

- **Site display name** - display name of the new site.
- **Site code name -** code name of the new site.
- **Domain name -** domain name on that the new site will be running.

Click **Next** to continue.



3. In Step 4, you can select which of the objects from the site package will be imported. You can make this selection by choosing one of the categories displayed in the tree on the left side of the screen. By selecting a category, a set of check boxes appears in the right part of the screen, letting you select which objects will to be imported.

If you select the root of the tree, you will be offered with the following options:

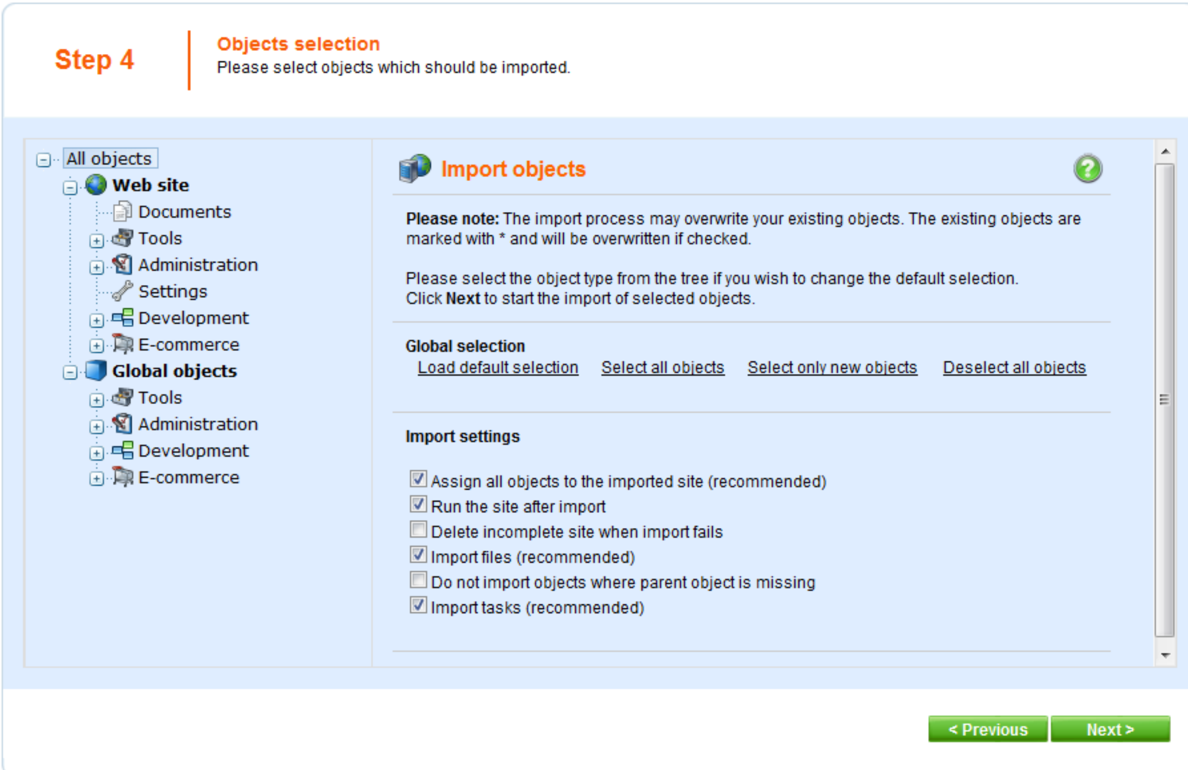
Global selection

- **Load default selection -** if clicked, object preselection will be done based on choice in Step 1.
- **Select all objects -** if clicked, all objects will be preselected.
- **Select only new objects** - if clicked, only objects not existing in the database will be preselected.
- **Deselect all objects** - if clicked, all objects will be deselected.

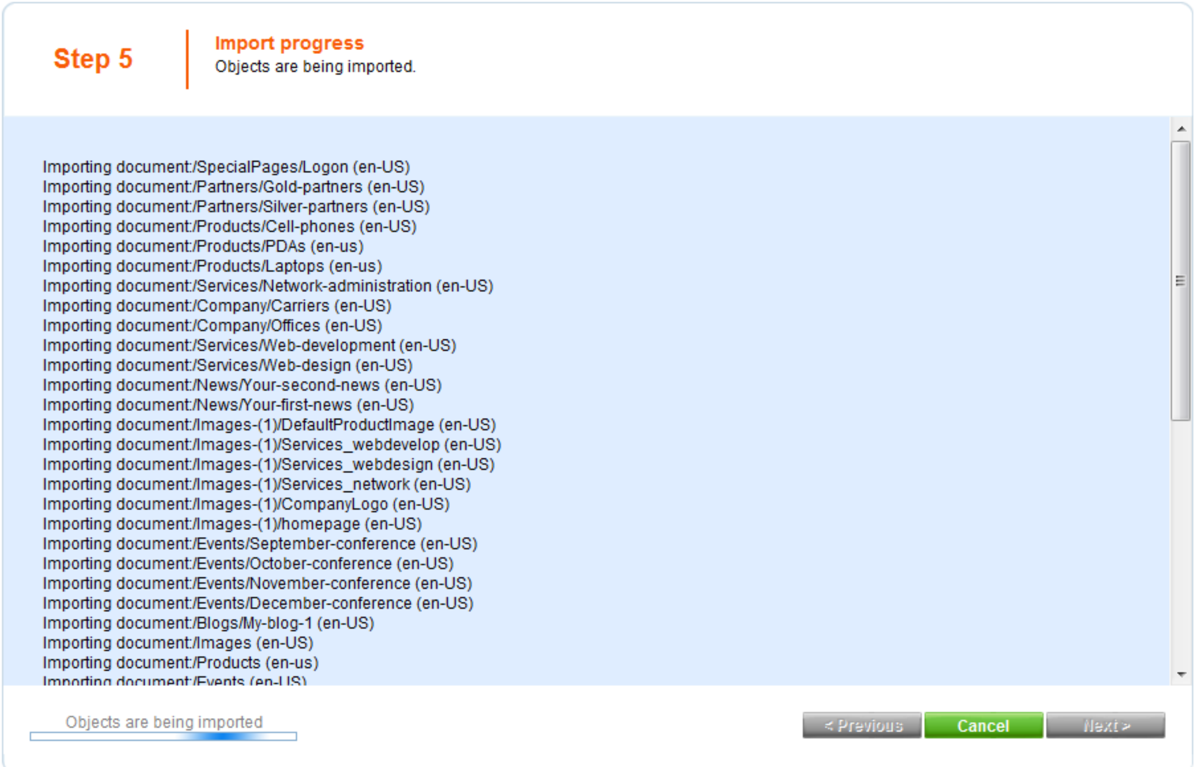
Import settings

- **Assign all objects to the imported site (recommended)** - if checked, all imported site related objects will be assigned to the imported site.
- **Run the site after import** - if checked, the updated site will be run after the import is finished.
- **Delete incomplete site when import fails** - if checked, incomplete site will be deleted when import fails.
- **Import files (recommended)** - if checked, files will be imported.
- **Do not import objects where parent object is missing** - if checked, objects that are in the package but whose parent object is not present in the target instance will not be imported
- **Import tasks (recommended)** - if checked, delete tasks (incremental deployment) included in the package will be imported

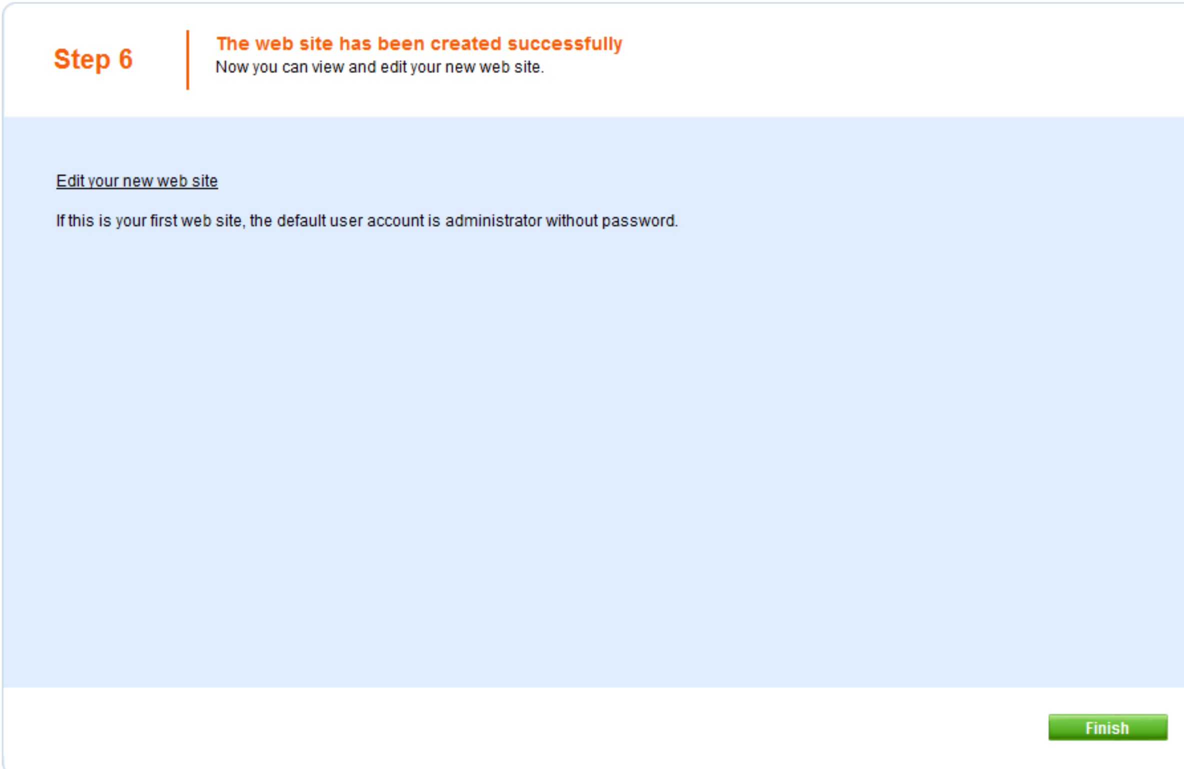
Click **Next** to continue.



4. A log will be displayed, showing you the progress of website import. When the export process finishes, click the **Next** button.



5. If you have reached Step 6, you have successfully created the new website. Click the **Edit your new website** link to switch to CMSDesk and start editing the site immediately. Alternatively, click the **Finish** button to get back to **Site manager -> Sites**.



3.3.5 Deployment to the live server

With Kentico CMS, you can easily develop the website on your local machine. When the website is ready to go live, you need to export it into an export package. You will need to make sure that the ASP.NET account has the **Modify permission** for the disk, specifically for the CMS*SiteUtils\Export* folder. See the Troubleshooting installation issues -> Disk permissions problems chapter for more information on how to set up these permissions.

Go to **Site Manager -> Sites** and click the **Export site** icon next to the site to be exported. Enter the name of the export package and click **OK**. The site will be exported to the **<web project>\CMSSiteUtils\Export** folder.

Now you need to install Kentico CMS on the live server. If you're using a remote server without desktop access, you need to:

1. Run Kentico CMS Web Installer **on your local development computer**.
2. Choose to create a new website **on a remote (production) server** and deploy the website into some local disk folder.
3. **Copy all deployed files** to the root of your web server (i.e. the web.config file must be placed in the root of the server). If you need to run the website in a sub-folder, you need to create virtual directory manually as described in Additional configuration tasks -> Creating a virtual directory.
4. Copy your exported package into the **<website root>\CMSSiteUtils\Import** folder.
5. Make sure your live server is **configured for ASP.NET 2.0**. It's also highly recommended that the ASP.NET account has Modify permission on the server disk for easy import (however, it's possible to complete the import without Modify permission as well).
6. Open the page /default.aspx of your live web server in the web browser.
7. You will be displayed with Database setup wizard. Create the database for Kentico CMS.

8. You may be asked for the license key for your production domain.
9. In step 3, choose to **Import existing Kentico CMS website**.
10. Choose to import the previously exported package and overwrite all duplicates. The import wizard is described in Managing sites -> Export and import -> Importing a site.

Now that you have imported the site, you may need to adjust its configuration:

1. Go to the **Sites** section. Edit website properties and make sure the website domain and domain aliases are configured correctly for the production domain(s).
2. Go to the **Site Manager -> Settings** and make sure your site settings contain correct values, especially the **SMTP server** value in the **E-mails** section.
3. Go to the **Sites** section. Click the **Open live site** icon next to your new site and make sure the website is displayed correctly.

Alternative Approach

If you, for some reason, cannot deploy the website using the procedure described above, you can use a classic backup/restore approach:

1. Backup your development database and restore it on the production server.
2. Copy the web project folder into the root of your production website.
3. Update the CMSConnectionString value in the web.config file.

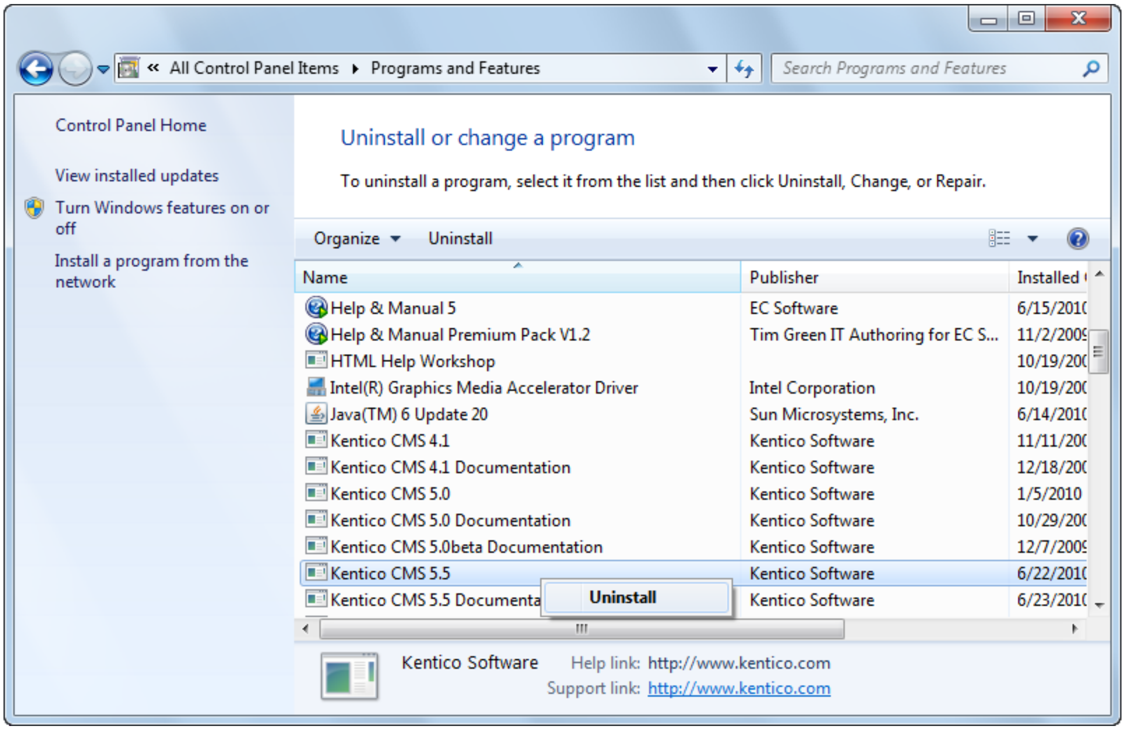
Shared Hosting Environment and Medium Trust Level

If you're deploying Kentico CMS on a shared hosting server that requires medium trust level, you may need to make additional configuration changes as described in Additional configuration tasks -> Configuration for Medium Trust environment.

Related topics: Visual Studio integration -> Pre-compilation (Publish function)

3.3.6 Uninstallation

To uninstall Kentico CMS from your computer, open **Start -> Control Panel -> Programs and Features**. Right-click **Kentico CMS** and choose **Uninstall** from the context menu.



The uninstaller will delete all files created on your computer except the deployed instances of Kentico CMS - it means that the database instances and the website folders will not be removed and you have to delete them manually if needed.

3.4 Additional configuration tasks

3.4.1 Overview

The following chapters contain description of various configuration tasks that require changes to your system. You needn't perform them for standard installation, they are required only in specific situations.

3.4.2 Creating a virtual directory

If you need to install Kentico CMS manually on a remote server or restore it from backup and, at the same time, you run Kentico CMS in a sub-folder (in contrast to running Kentico CMS in the root of the website), you need to create a new virtual directory for the folder where you have the web project files.

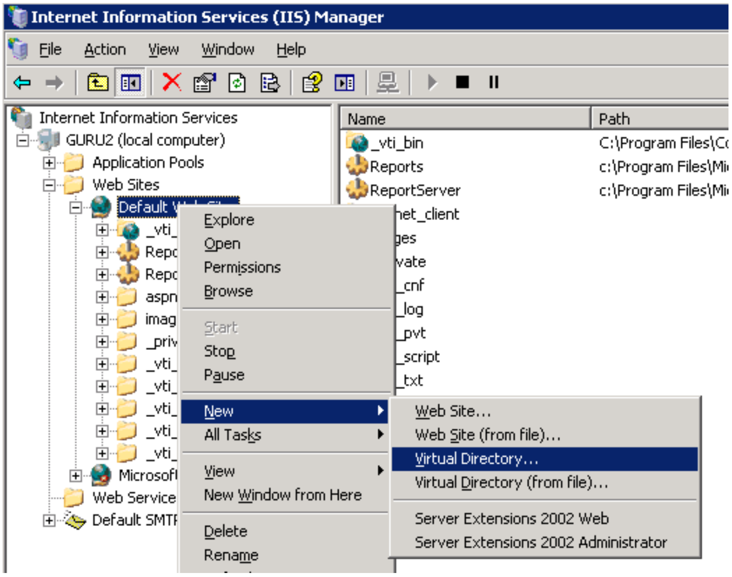
Application root

The root of the website or the virtual directory must be the same as the folder that contains the web.config file of Kentico CMS. This folder is called **application root**.

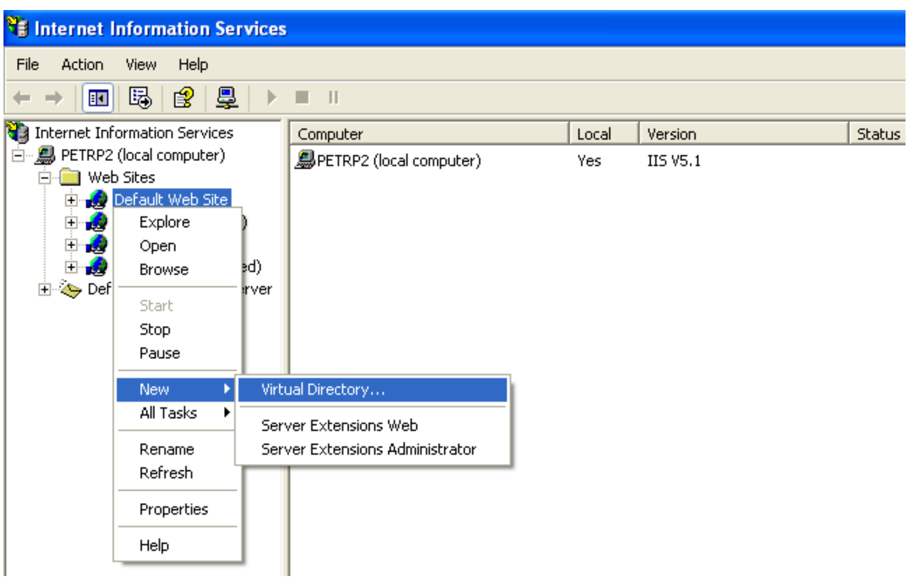
Follow these steps to create and configure a virtual directory on your web server:

Configuring a Virtual Directory

On Windows Server 2003, open **Start -> Administrative Tools -> Internet Information Services (IIS) Manager**. Right-click **Computer -> websites -> chosen website** and choose **New -> Virtual Directory**.



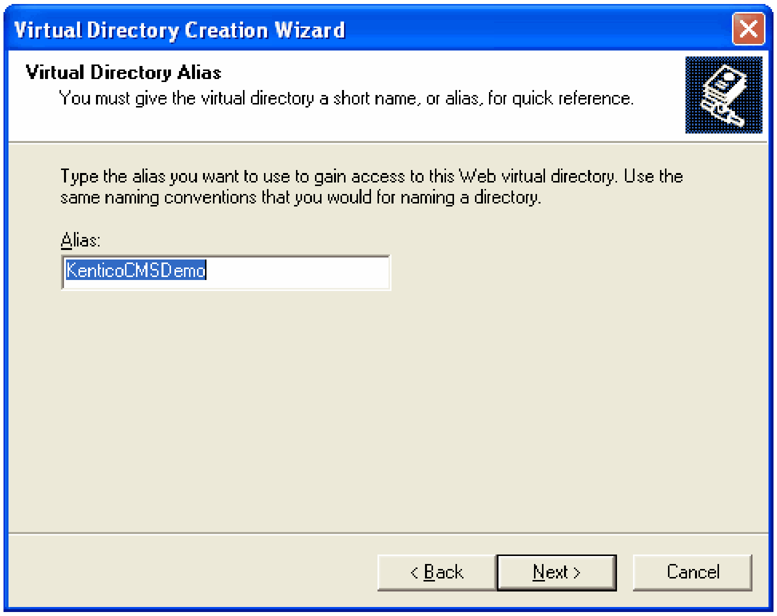
On Windows XP, **open Start -> Control Panel -> Administrative Tools -> Internet Information Services**.Right-click the **Computer -> websites -> Default website** item (or other website if you're running multiple websites on the same computer) and choose **New -> Virtual Directory...**.



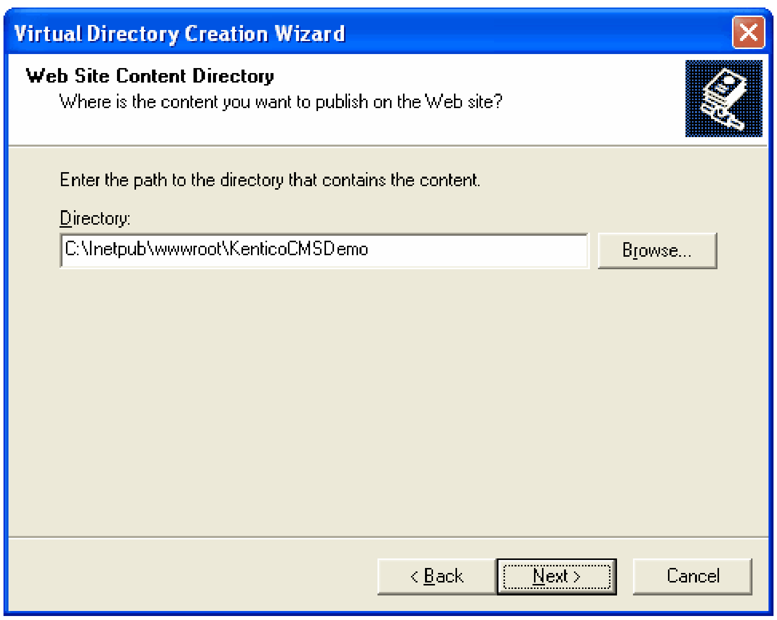
The rest is same for all operating systems:

1. The **Virtual Directory Creation Wizard** appears. Click **Next** on the first screen.

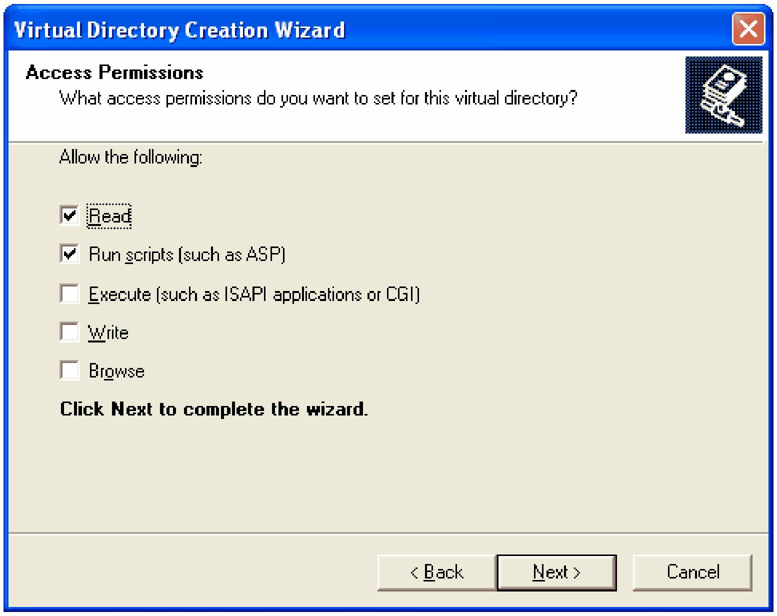
2. Enter the **Virtual Directory Alias**. If you want the website to run as http://localhost/KenticoCMSDemo, you need to enter alias **KenticoCMSDemo**. Click **Next**.



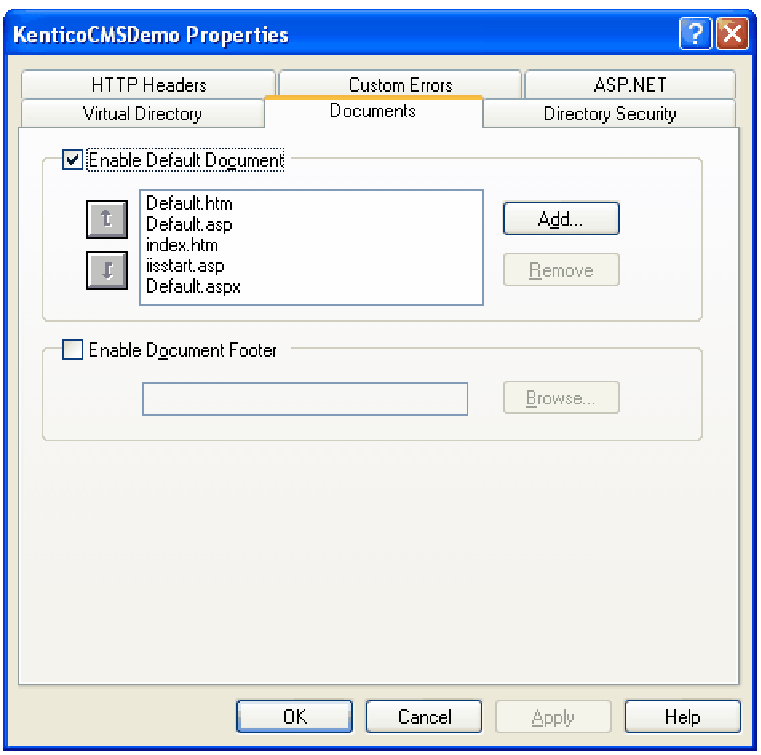
3. Now choose the disk folder on your machine where the Kentico CMS web project files are placed. Please note that you cannot use a remote (shared) disk. The web project files are NOT the files in the C:\Program Files folder - you need to create the web project using the Kentico CMS Web Installer first - see chapter Web Installer for details. Enter the directory path and click **Next**.



4. Now you need to specify the access permissions for the new virtual directory. You need to choose at least **Read** and **Run scripts** and it's recommended that you do NOT allow any other options. Click **Next** and click **Finish**.



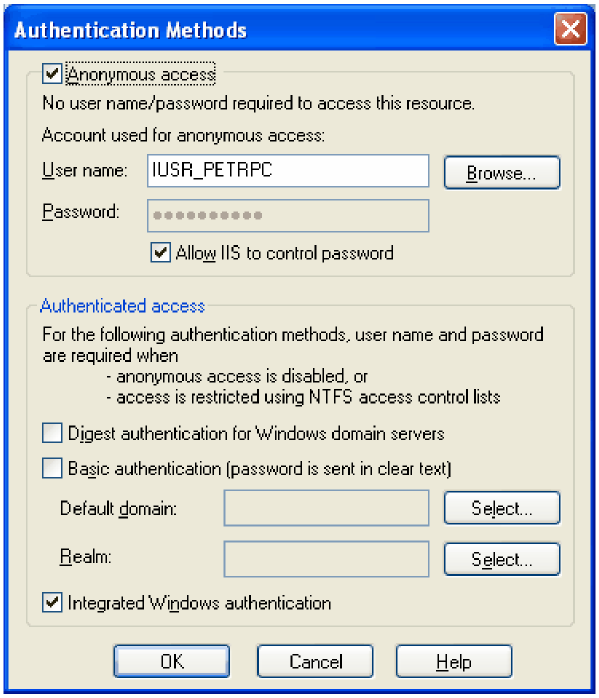
5. Right-click the newly created virtual directory and choose **Properties**. Choose the **Documents** tab and make sure the **Enable Default Document** box is checked and the list contains the **Default.aspx** file.



6. Choose the **ASP.NET** tab and make sure the **ASP.NET version is 2.0.50727** (the last number may be different). If you cannot see the **ASP.NET** tab, you may not have **ASP.NET 2.0** installed or registered correctly.

7. Choose the **Directory Security** tab, click **Edit** in the **Anonymous access and authentication**

control section and make sure that both **Anonymous access** and **Integrated Windows authentication** are chosen.



8. Click **OK** on all dialogs to save the changes.

3.4.3 Configuration for Medium Trust environment

This chapter describes the procedure of running Kentico CMS in Medium Trust Level environment and the steps required to configure the system for it. It also describes the Precompilation/Deployment procedure and requirements.

Medium trust level

Medium trust level is often used on shared servers by the providers to disallow the applications to access certain resources that could be harmful to other websites running on the server. Kentico CMS can be used with default medium trust policy provided with the Microsoft .NET Framework. To run the system with medium trust, you need to follow certain rules. There are two main components that require higher than medium trust and must be considered in this situation:

- **VirtualPathProvider** – provides virtual objects (layouts, transformations) from the database.
- Content staging module – ensures synchronization of content between production and live site servers.

Virtual path provider

This library provides the interface for virtual objects stored in the database that can be compiled, such as document transformations and page layouts. The system references the files with virtual path, and VirtualPathProvider provides the control code to the compiler.

Since the virtual provider cannot run in a medium trust environment (requires AspNetHostingPermission with "high" trust level), you need to store the physical files to the file system. You can save all the virtual objects to the file system in **CMS Site Manager -> Administration -> System -> Deployment interface by clicking the button "Save all virtual objects to disk"**. This will make the copies of the virtual objects in following folders:

- **~/CMSTransformations** - contains transformation files for documents and custom tables
- **~/CMSLayouts** - contains shared page layouts
- **~/CMSTemplateLayouts** - contains custom page template layouts
- **~/CMSAdhocTemplateLayouts** - contains custom ad-hoc page template layouts

Please note that these files are just copies of the actual virtual object and will be used by the system only if the VirtualPathProvider cannot start. Also that the changes to the objects through the administration interface will not affect these files until you save all the objects to the disk again.

Limitations

In the medium trust environment, the VirtualPathProvider is stopped automatically. When VirtualPathProvider is stopped, **you cannot edit transformations and layouts through the user interface** without saving them on the disk again.

In the portal development model, you cannot use **custom web part code** (**Web part properties –> Code tab**). If you need to add custom code on the Code tab and run the website in the medium trust environment, you need to create user controls, place web parts to the user controls and add your custom code to the web parts. Then, you can place the user controls to the page using the General/User control web part.

Please note that **you shouldn't run the system in medium trust while developing the website**. You should use this trust level only for the live website.

Content staging (Microsoft Web Services Extensions 3.0)

This section applies only if you're using content staging.

Library Microsoft.Web.Services3.dll from the Web Services Extensions 3 (WSE) package which is used by the Content staging module requires Full trust permissions because of the low level operations related to the communication protocols. To ensure the proper functionality, the library needs to be registered in the Global Assembly Cache (GAC) of the server. The library is provided by Microsoft and most of the hosting providers pre-install it on their shared servers.

If you manage the server, please follow these steps:

1. Go to **Control panel -> Administrative tools -> Microsoft .NET Framework 2.0 Configuration**.
2. Select the **Assembly cache**, click on **Add an Assembly to the Assembly Cache** and select the *bin\Microsoft.Web.Services3.dll* library file from your web project.
3. Delete the *bin\Microsoft.Web.Services3.dll* file from your web project if it's present.
4. Make sure that your project *web.config* file contains the following item:

```
<system.web>
  ...
  <compilation debug="false" numRecompilesBeforeAppRestart="100">
    <assemblies>
      ...
<add assembly="Microsoft.Web.Services3, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
...
    </assemblies>
  </compilation>
  ...
</system.web>
```

If your hosting provider cannot install the library to GAC, you need to manually remove some of the system components. See the paragraph **Special cases** for details.

Running the website

Now the system should work under medium trust level properly. Restart your IIS for configuration changes to take effect and run the website.

If your website uses any third-party components that do not support medium trust level by default, you may need to configure the system for them. In this case, please contact their author to get the information how to configure the environment to run it in the medium trust environment.

Special cases

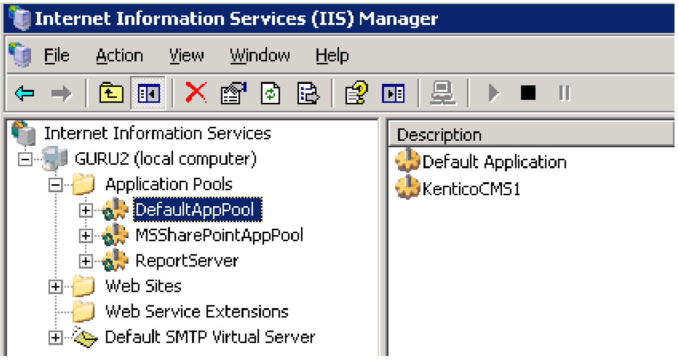
You may not be able to access the GAC or convince your hosting provider to add the WSE library to the GAC. In this case, you may still run Kentico CMS in medium trust level, but you will not be able to use the Content staging module.

If this is your case, please delete the *bin/Microsoft.Web.Services3.dll* file from your web project if it's present

After these changes, your system will work correctly in the medium trust environment but you will not be able to use the Content staging operations.

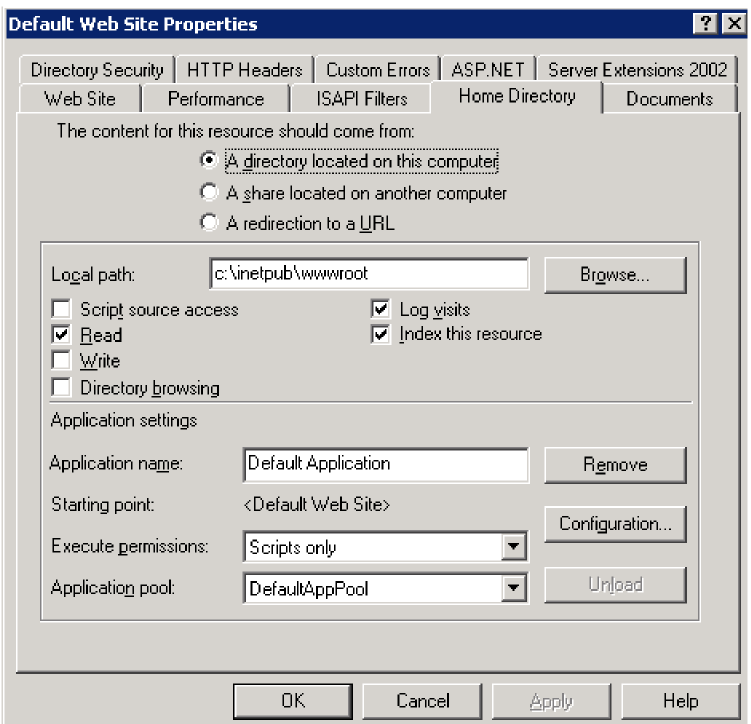
3.4.4 Configuring Application Pools

Application Pools provide you with additional level of website management. They are supported only on Windows Server 2003/2008/Vista/7. You can configure them in **Start -> Control Panel -> Administrative Tools -> Internet Information Services (IIS) Manager -> Computer -> Application Pools**.

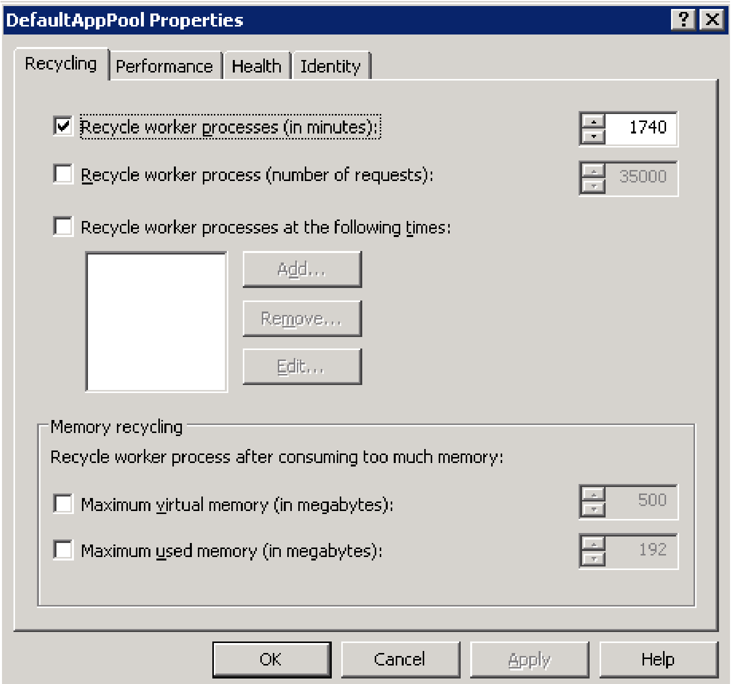


The application pools allow you to group applications (websites) into pools that share server resources using the same rules. This chapter contains recommendations on how to configure the website application pool for Kentico CMS.

You can check the name of the application pool in the website Properties dialog, on the Home directory tab. If you're running Kentico CMS in a virtual directory, you will find it in the Virtual Directory Properties dialog, on the Virtual Directory tab.



You can see the **Application Pool Properties** dialog by right-clicking the appropriate application pool and selecting Properties from the context menu. You will see dialog like this:



Recommended Application Pool Configuration

1. It's highly recommended that you run Kentico CMS in a separate application pool. If you share the pool with other websites, the system may behave unpredictably.

2. If you need to run multiple websites in a single pool, be sure to run only ASP.NET 2.0 applications in the same pool.

3. It's recommended that you specify some value in the **Recycle worker** processes on the **Recycling** tab.This value shouldn't be too short (less than **60 minutes**) or too long (more than **1440 minutes**/1 day). Setting this value ensures that the **memory is recycled and the application is automatically recovered from failures** by regular restart. **If your website freezes** time to time, you can temporarily set the value to 30 minutes to ensure that the website is restarted automatically. **Short intervals** may lead to high server load and slow response since after each recycling, the application needs to be restarted and data reloaded to the cache.

4. It's recommended that you **do not limit the maximum virtual or used memory**. If you need to use some value, use **at least 100 MB**. If your website is being **restarted too often**, it may be caused by low maximum memory limit. You can check the frequency of application restarts in Kentico CMS Event Log (Site Manager -> Administration -> Event log).

5. The **Maximum number of worker processes** on the **Performance** tab must be set to 1. If you set a higher value, the worker processes will not be synchronized and Kentico CMS website will not work correctly. This may lead to **unexpected caching of content and system objects**.

6. You can configure the **user account** under which the application runs on the **Identity** tab. This information is useful if you need to troubleshoot issues with **permissions**, such as disk write permissions.

7. Kentico CMS **does not support Web garden**. Therefore, the **Maximum number of worker**

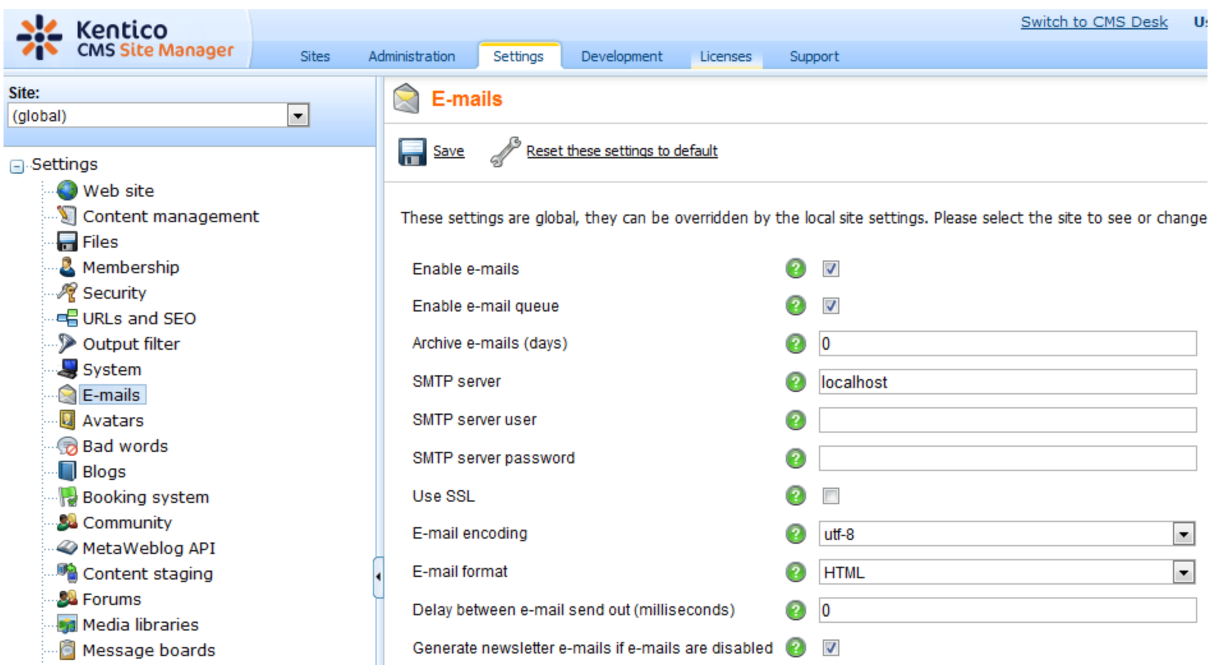
processes has to be set to **1**.

3.4.5 SMTP server configuration

If you want to send e-mails from the web application, you may need to configure the SMTP server.

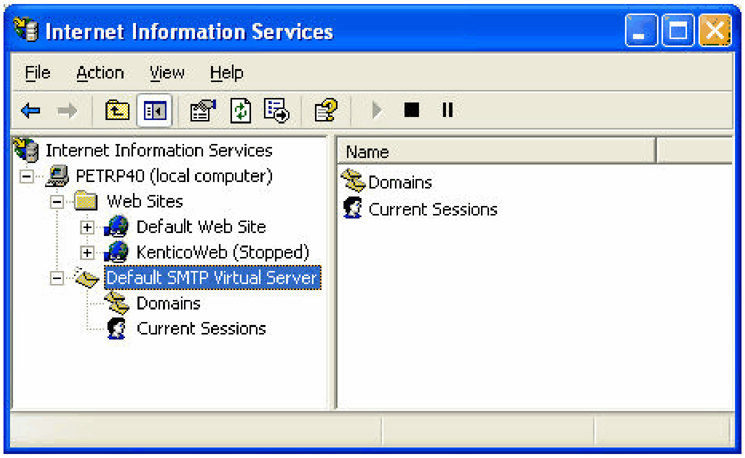
After you finish the New site wizard, you can configure the SMTP server in **Site Manager -> Settings -> E-mails**. Here, you can set up the following values:

- **SMTP server** - The name, address or IP address of your SMTP server, including the port number (if it's not 25).
- **SMTP server user** - User name - use this value only if your SMTP server requires authentication.
- **SMTP server password** - Password - use this value only if your SMTP server requires authentication.
- **E-mail encoding** - Here you can enter the encoding of the e-mails. The default and recommended value is UTF-8. Change this value only if you encounter issues with extended characters in your e-mails.

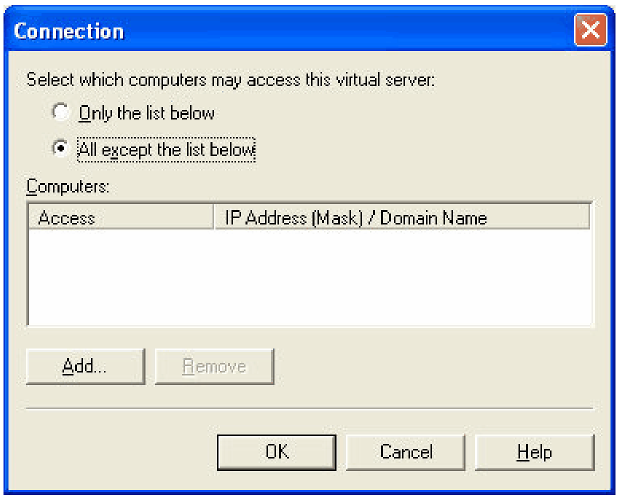


If you experience any problem when sending e-mails through your own SMTP server that is part of IIS, please follow this guide:

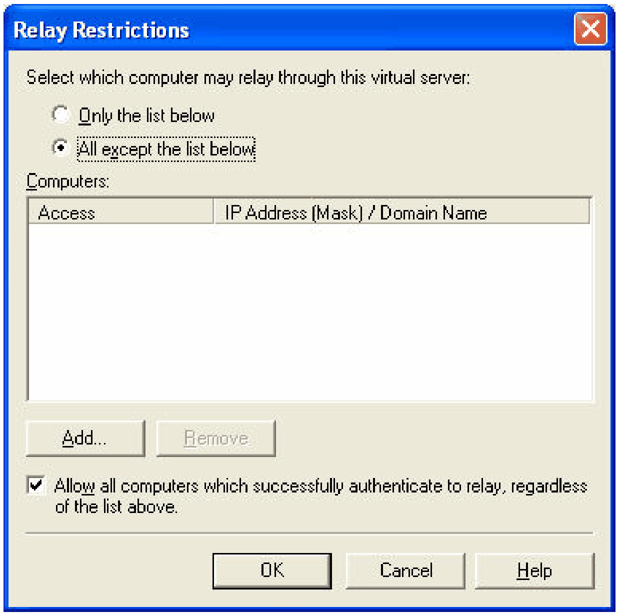
1. Open **Control Panels -> Administrative Tools -> Internet Information Services** console. Check that the SMTP server is installed and running:



2. Open SMTP server properties and choose the **Access** tab. Click **Connection control** and make sure that the local machine may access the SMTP server:



3. Click **Relay restrictions** and check that the local machine can relay through the SMTP server:



Testing E-mail Sending

You can test sending e-mails from Kentico CMS using the **CMS Site Manager -> Administration -> System -> E-mail dialog**.

3.4.6 Installation on shared hosting server

If you want to deploy Kentico CMS directly to the live server (or generally to a remote server), you need to follow these steps:

1. Install Kentico CMS on your **local machine** if you haven't done so yet.
2. Run Kentico CMS Web Installer (you can find it in Start -> All Programs -> Kentico CMS).
3. Choose to **install Kentico CMS on a remote (production or testing) server** and enter the path *C:\tempfiles*. Finish the wizard - it only copies the files to the given folder. You do not need to have Visual Studio installed.
4. **Copy the files** from your local folder *C:\tempfiles* to the root of the website using FTP. If you want to use a sub-folder, you will need to create a new virtual directory as described in Creating a virtual directory.
5. **Open web browser** and navigate to your website. You will be displayed with **Database Setup Wizard**.
6. Go through the wizard and create a **new Kentico CMS database** on your live server.
7. At the end of the Database Setup Wizard, choose either to create a new website or import your existing Kentico CMS website.

Installation in Medium Trust Level

If you're installing Kentico CMS in medium-trust environment, please read chapters

Installation_in_medium-trust_environment and Configuration_for_Medium_Trust environment.

Related topics: Deployment to the live server, Configuration for Medium Trust environment

3.4.7 Installation in medium-trust environment

If you want to deploy Kentico CMS directly to the live server (or generally to a remote server) that uses **medium-trust security level**, you need to follow these steps:

1. Install Kentico CMS on your **local machine** if you haven't done so yet.
2. Run Kentico CMS Web Installer (you can find it in Start -> All Programs -> Kentico CMS).
3. Choose to **install Kentico CMS on a remote (production or testing) server** and enter the path C:\tempsite. Finish the wizard - it only copies the files to the given folder. You do not need to have Visual Studio installed.
4. **Copy the files** from your local folder C:\tempfiles to the root of the website using FTP. If you want to use a sub-folder, you will need to create a new virtual directory as described in Creating a virtual directory.
5. Make sure the web.config file on your server contains the following value in the **appSettings** section (it specifies that the CMS should use the managed provider):

```
<add key="CMSDirectoryProviderAssembly" value="CMS.DirectoryProviderDotNet" />
```

6. **Open web browser** and navigate to your website. You will be displayed with the first step of the Database Setup.
7. Go through the wizard and create a **new Kentico CMS database** on your live server. At the end of the process, you will be asked to update your web.config file manually - please follow the instructions on the screen.
8. Choose either to create a **new website or import your existing** Kentico CMS website.
9. After you create a new website, you will need to save all virtual objects to the disk. Go to **Site Manager -> Administration -> System -> Deployment** and click **Save all virtual objects to disk**.

If your web application is not allowed to write on the disk, you will need to install Kentico CMS on your local machine, save all virtual objects to disk and then copy the web project over FTP to your server.

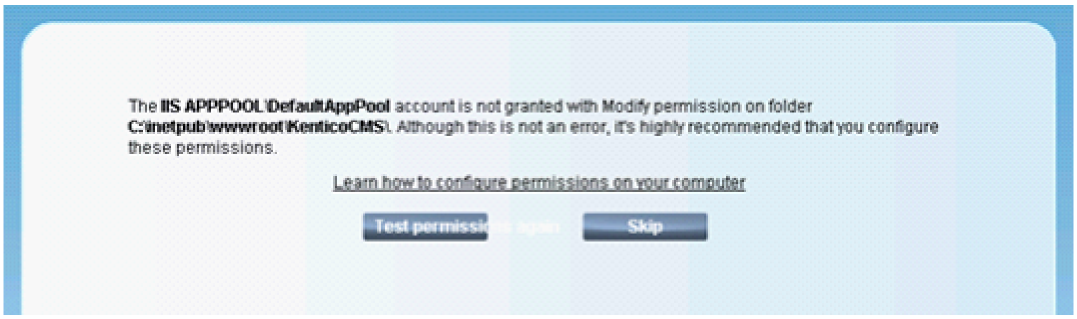
Medium Trust Environment Specifics

For more information on medium trust level please read chapter Configuration for Medium Trust environment.

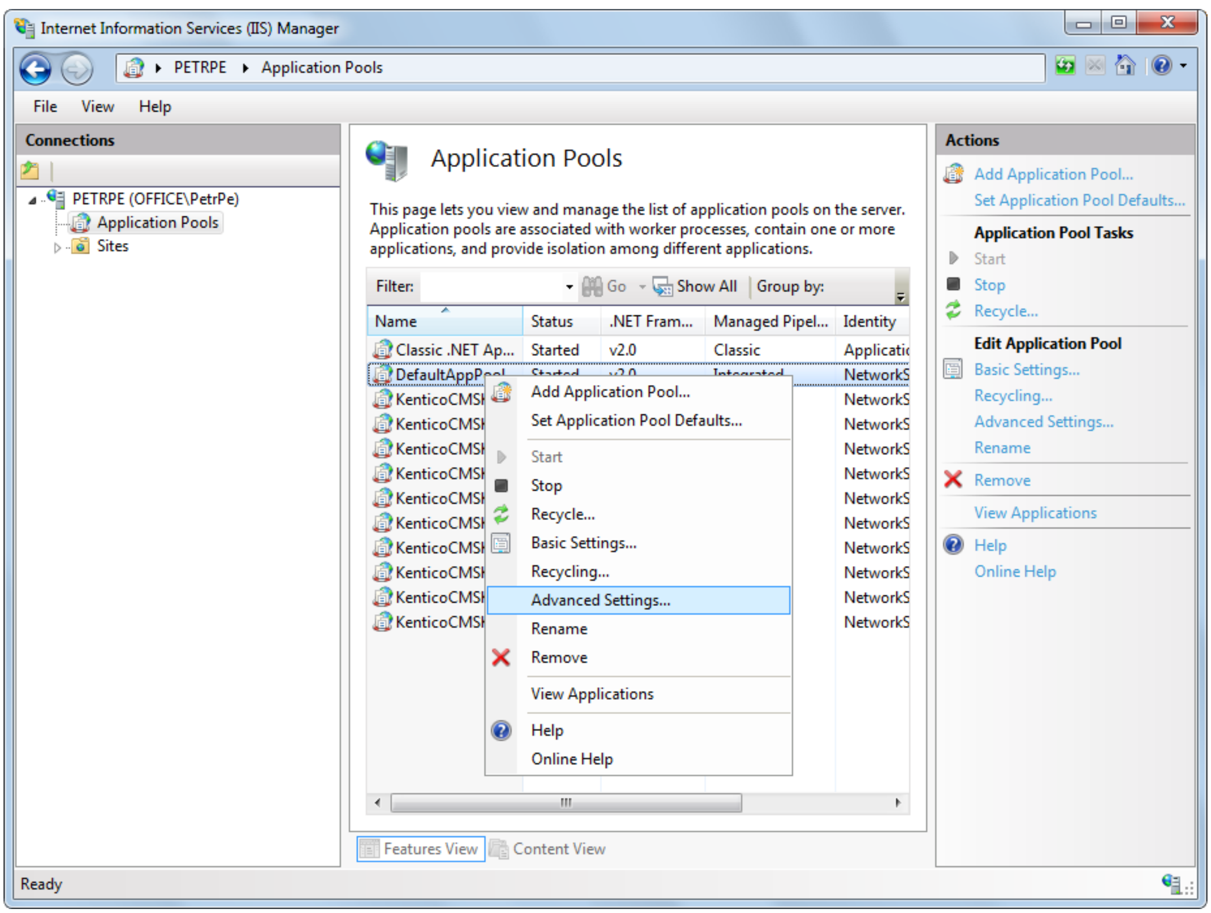
3.4.8 AppPool permissions on Windows 7 or Windows Server 2008 R2

Kentico CMS runs on both **Windows 7** and **Windows Server 2008 R2**. On both systems, you need to add the **Write** permission to the website's folder as described in Troubleshooting installation issues -> Disk permissions problems.

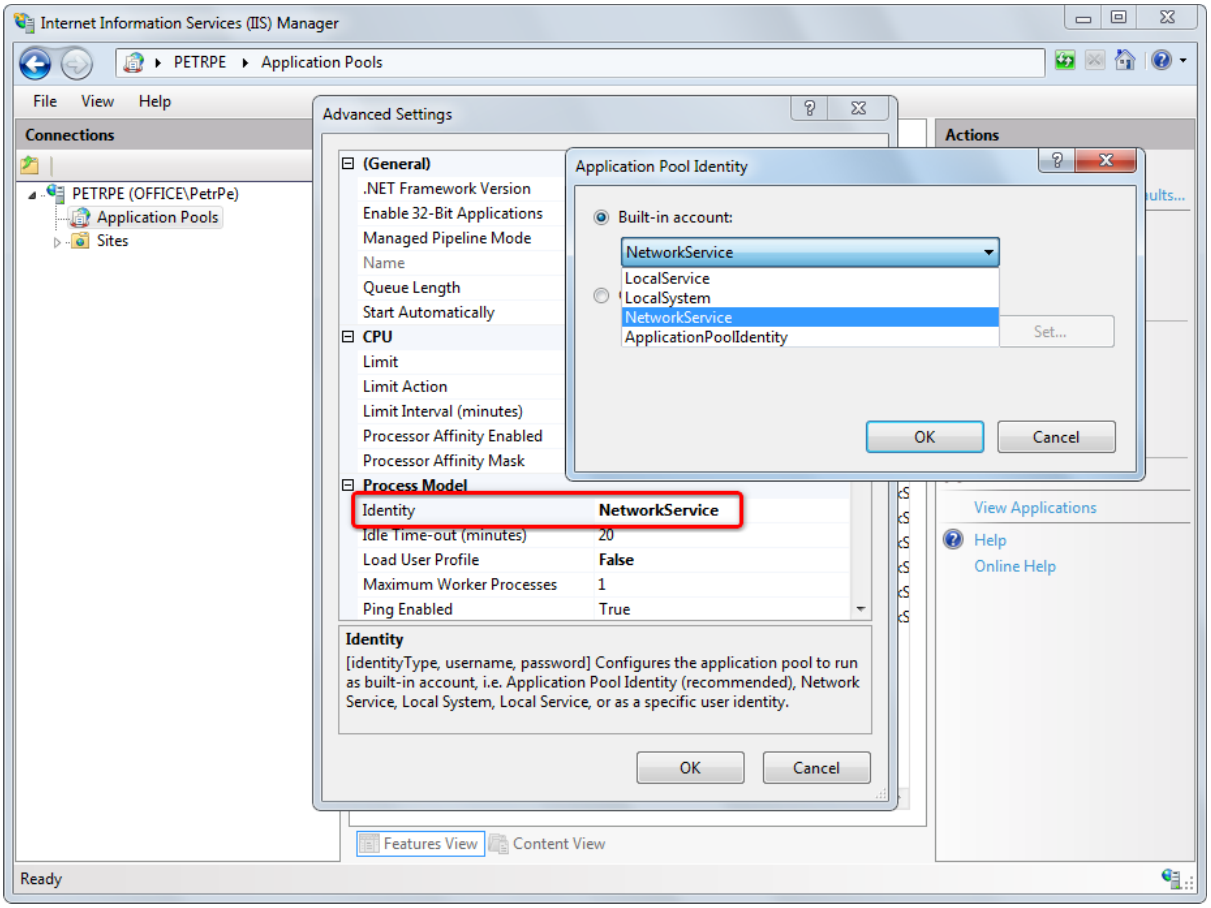
If you can still see the following screen after the end of the Database setup in either of the systems, it's necessary to make some further settings in your IIS.



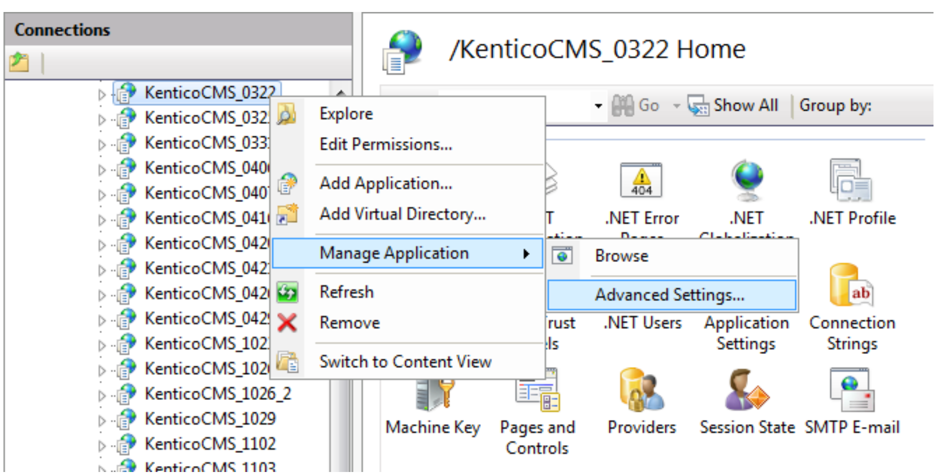
1. Select the **Application Pools** node in the **Connections** tree in the IIS console. All application pools will be listed. Right-click the **DefaultAppPool** and chose **Advanced settings** from the context menu.



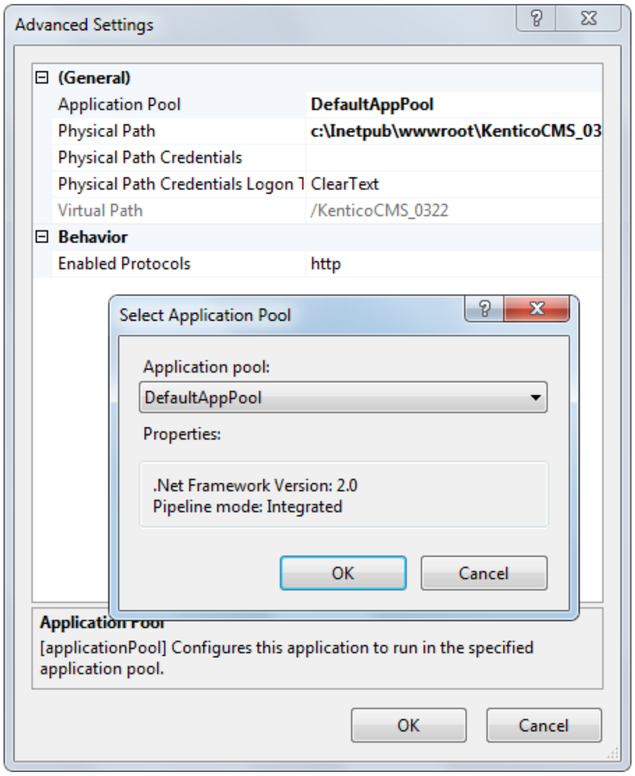
2. In the following dialog, switch the **Identity** value in the **Process Model** category to **Network Service**.



3. Now go to your website's **Advanced settings** ...

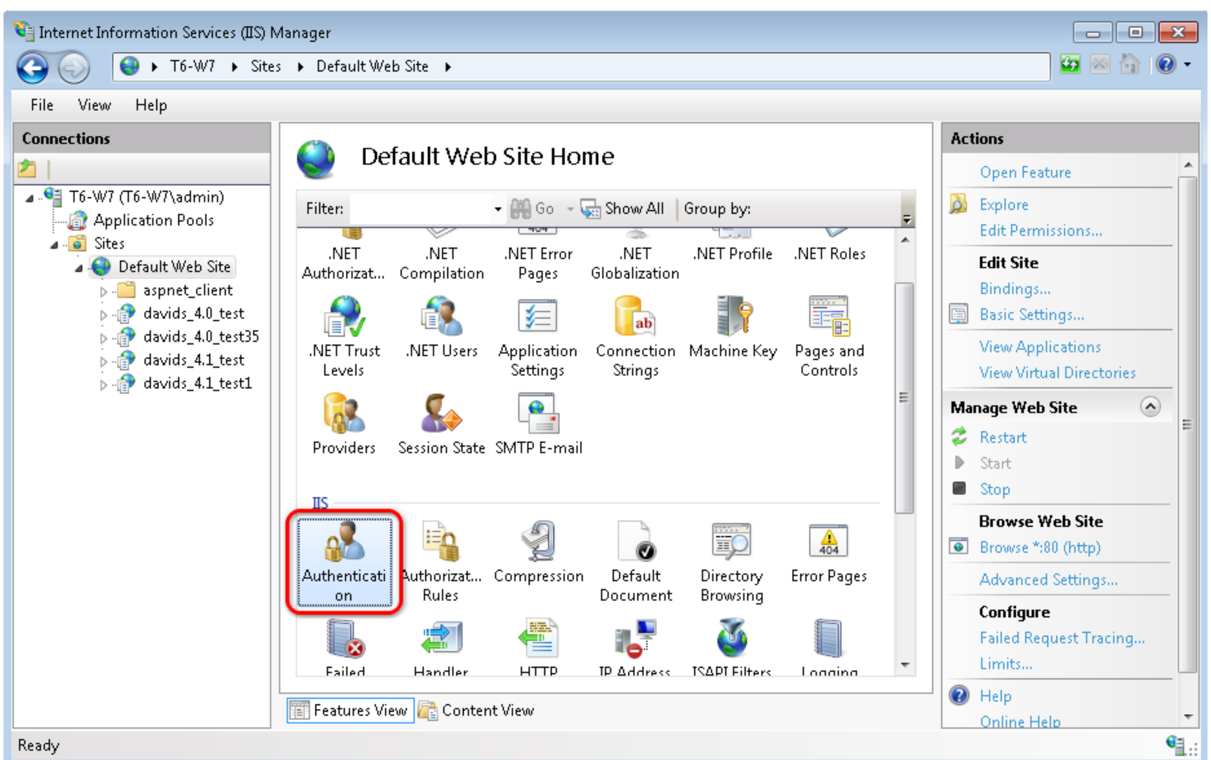


4. ... and set the **Application Pool** property to **DefaultAppPool**.

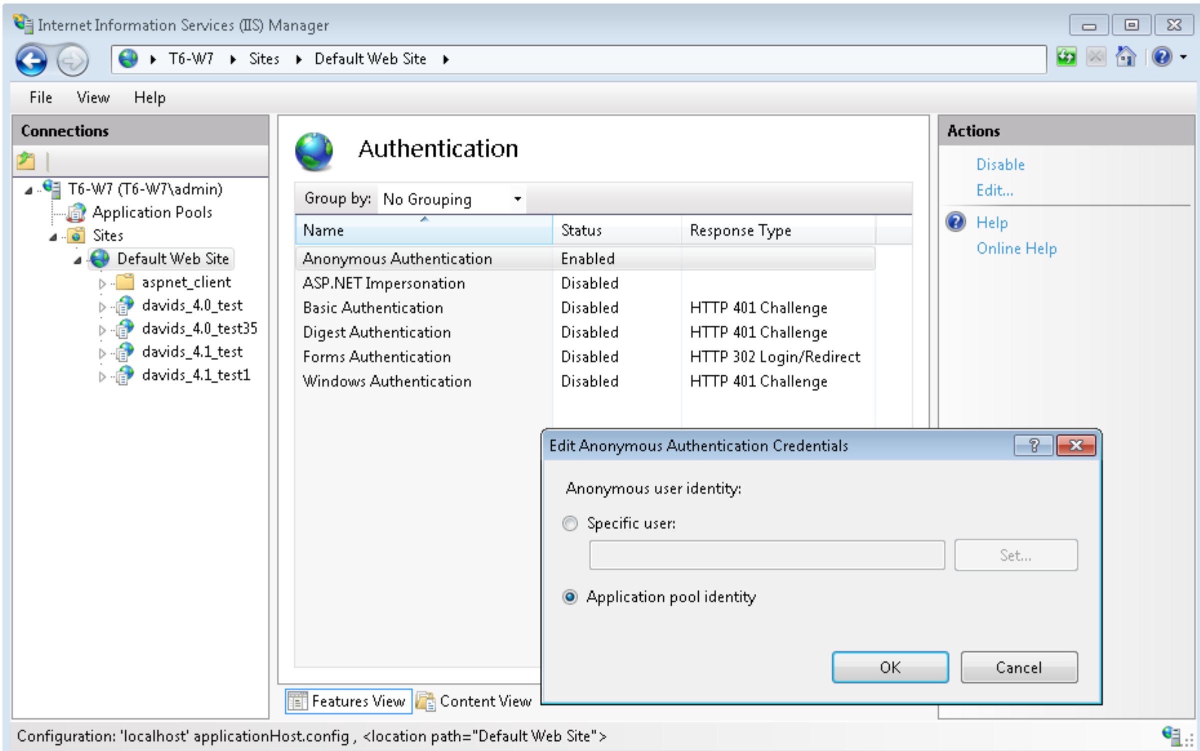


On **Windows 7**, you also need to change the Anonymous authentication settings as described below:

1. Open **IIS manager**, select your site and double-click the **IIS -> Authentication** icon.



2. Select **Anonymous authentication** from the list and click **Edit**. In the pop-up dialog, choose **Application pool identity** and click **OK**.



3.4.9 Database replication

3.4.9.1 Overview

Replication is a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency. The databases can be running either on separate servers or even on the same server. You will typically use it for back-up purposes (you have two databases with the same content) or in case that you need to spread the load among more databases.

Kentico CMS database can be replicated using the **merge replication**. It can be configured in MS SQL Server Management Studio. The configuration process consists of creating a publication and one or more subscriptions:

- Publication - the database which has the data to offer to the other server
- Subscription - the database which receives updates from the publisher when the data is modified

Detailed information about replication in MS SQL can be found at: http://msdn.microsoft.com/en-us/library/ms151198.aspx.

Limitations caused by DB replication!

Using database replication with Kentico CMS results in the following limitations:

- importing is not functional
- upgrade from previous versions of Kentico CMS to a higher one is not possible
- newly created document types, BizForms and custom tables can not be synchronized to the subscribers
- field changes in document types and system tables can not be synchronized to the subscribers
- any other additional changes to the database structure are not guaranteed to be synchronized to the subscribers

It is therefore recommended to go through the following procedure when your website is complete, i.e. when you won't need to use any of the limited features.

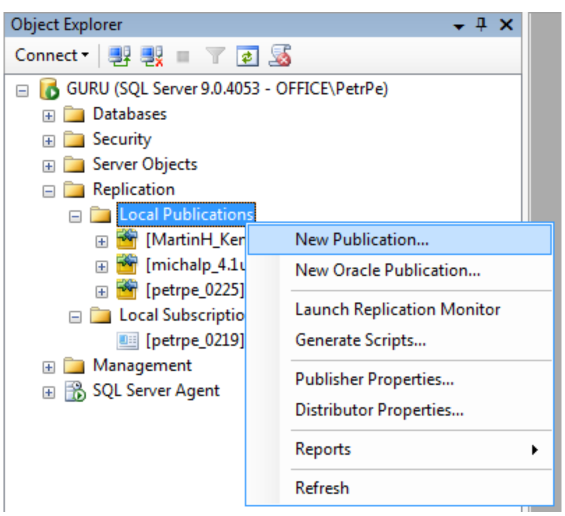
In case that you needed to use any of the limited features listed above, you would need to disestablish your existing replication, make the required changes and set up new database replication from scratch.

If you want to achieve this manually without disestablishing the existing replication, you can follow the instructions on this page.

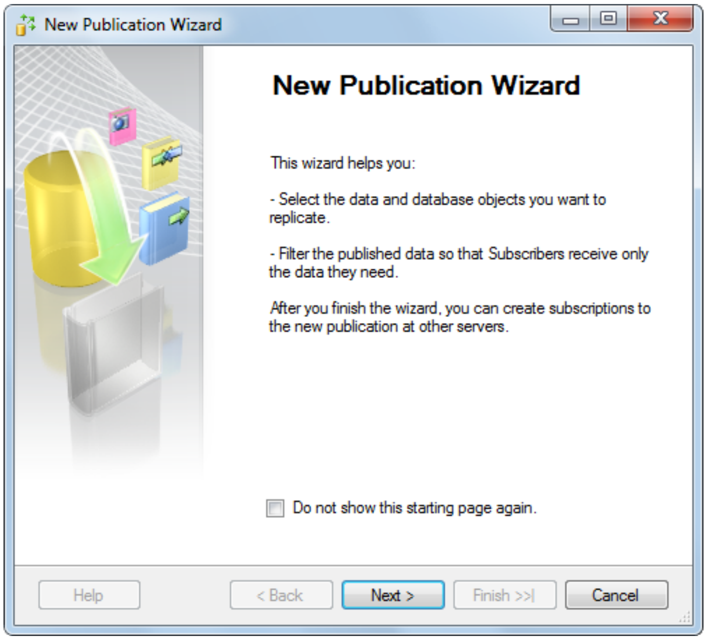
3.4.9.2 Creating a publication

The first logical step in setting up database replication is to create the publication.

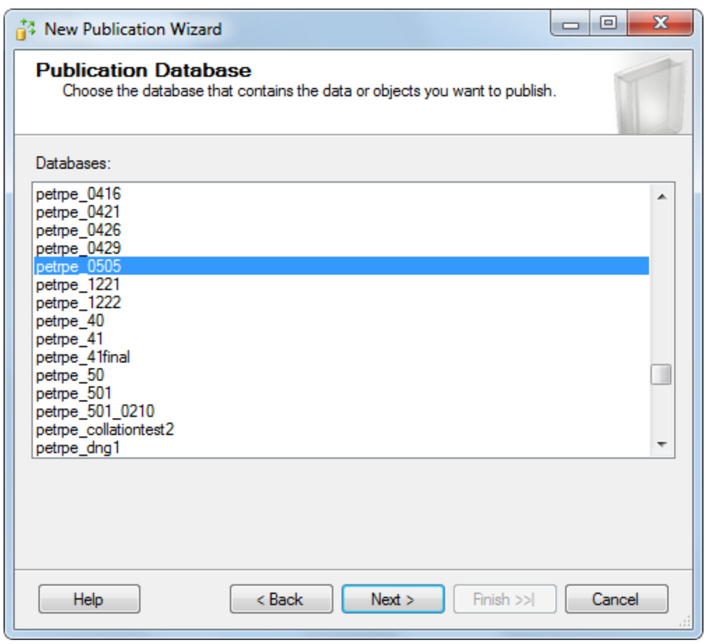
1. Open MS SQL Server Management Studio. Expand the **Replication** node, right-click the **Local Publications** folder and select **New Publication** from the context menu.



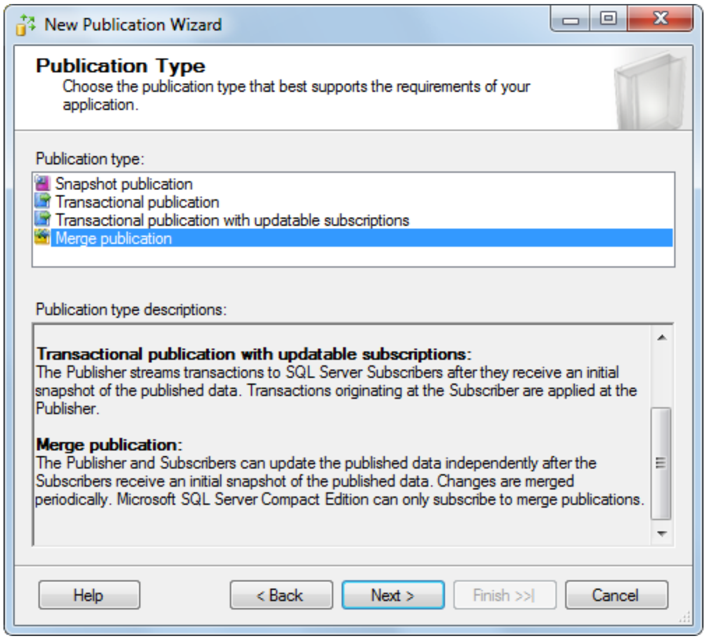
2. The **New Publication Wizard** starts. Click **Next** in the first step.



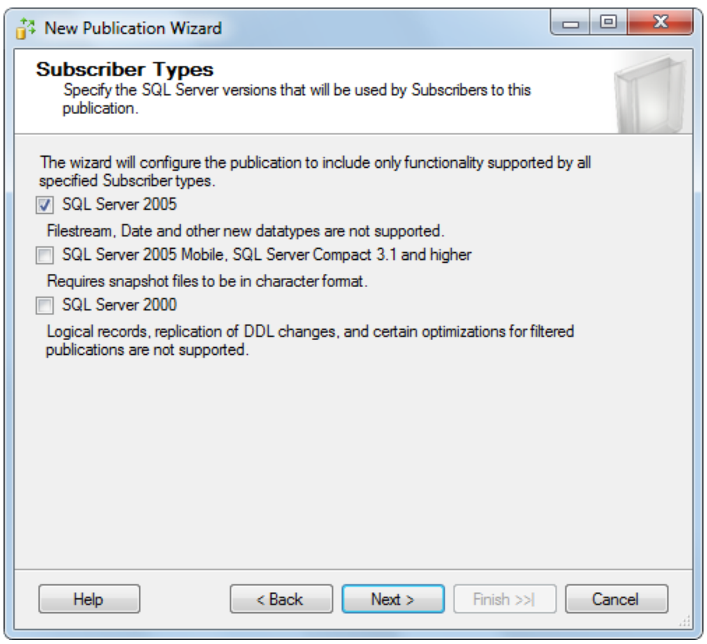
3. Now you need to select the publication database. All databases on the server will be listed, so choose the one which you want to be the publisher and click **Next**.



4. The **Publication Type** step lets you choose the type of replication which you are going to use. Kentico CMS supports merge replication only, so choose **Merge publication** and click **Next**.

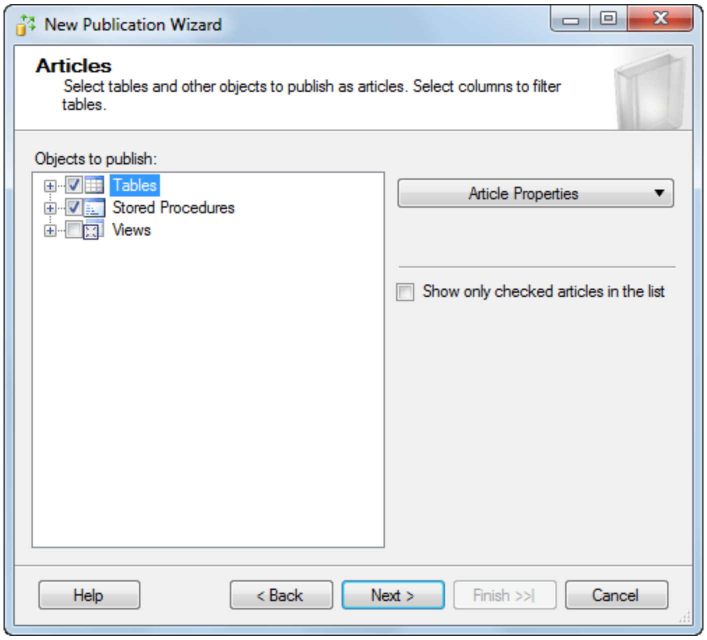


5. This step lets you specify which versions of SQL Server are supported for the subscribers. Choose the versions that your subscribers are running on and click **Next**.



6. This step lets you configure which **Tables** and **Stored Procedures** will be synchronized between the servers. You should not select any **Views** as because changes in metadata which cause changes in any of the Views can not be synchronized automatically (see the note in Overview for more details).

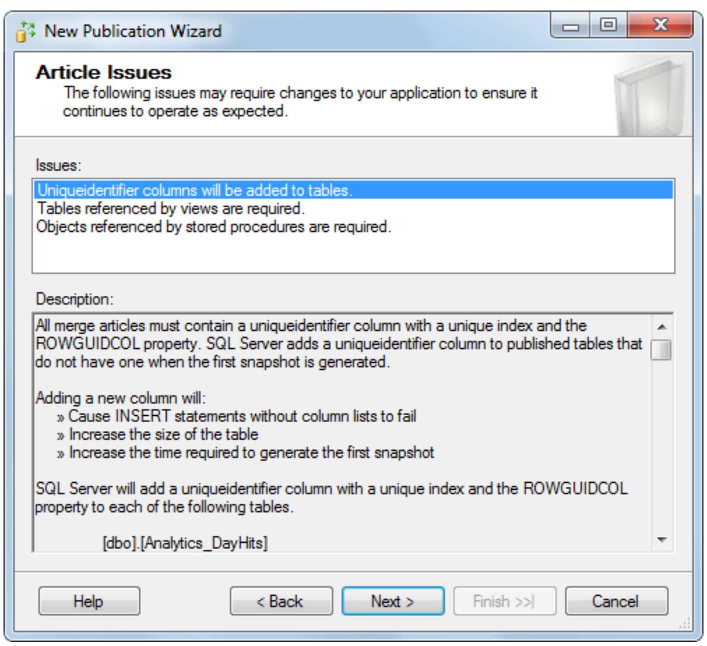
Select the tables and stored procedures that you wish to synchronize (unless you have some specific needs, it makes the most sense to select all) and click **Next**.



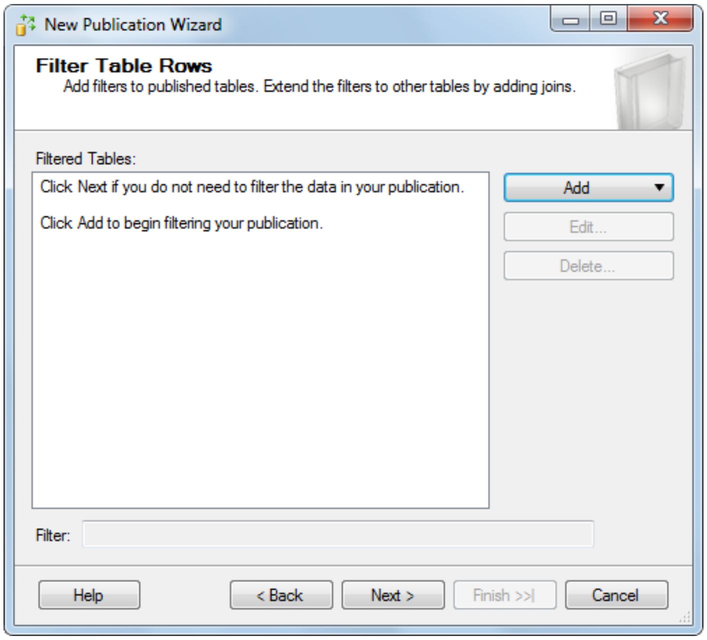
7. The **Arcile Issues** step is only informational - it notifies you about modifications to your database which are required for the application to work as expected with the replication.

The most important modification is that a column named **rowguid** (uniqueidentifier column with a unique index) and the **ROWGUIDCOL** property are added to each table.

Read through the info and click **Next**.



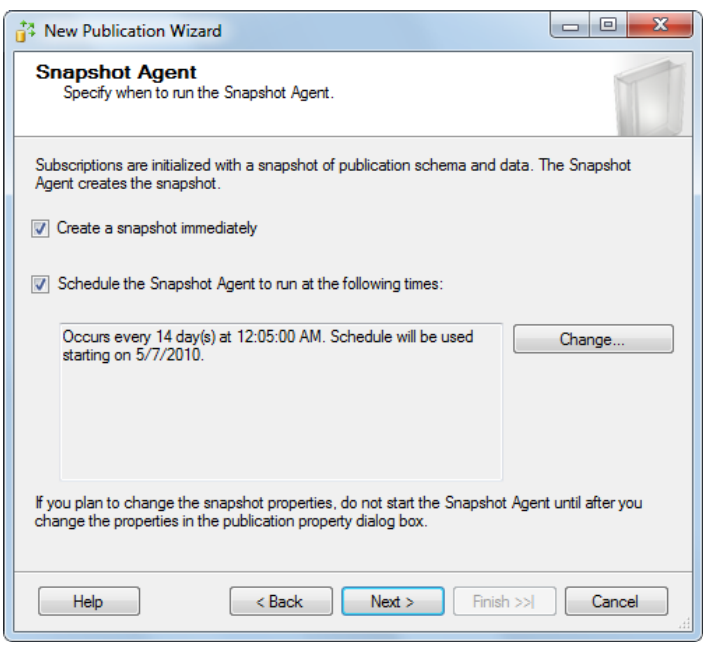
8. In the **Filter Table Rows** step, you can add filters the published tables. This depends on your specific needs, so add filters as needed (or do not add any) and click **Next**.



9. Now you need to configure the **Snapshot Agent** to take a snapshot of the publisher database. The snapshot will be used for the initial content of the subscriber database, which can not be created without it. You have the following two options:

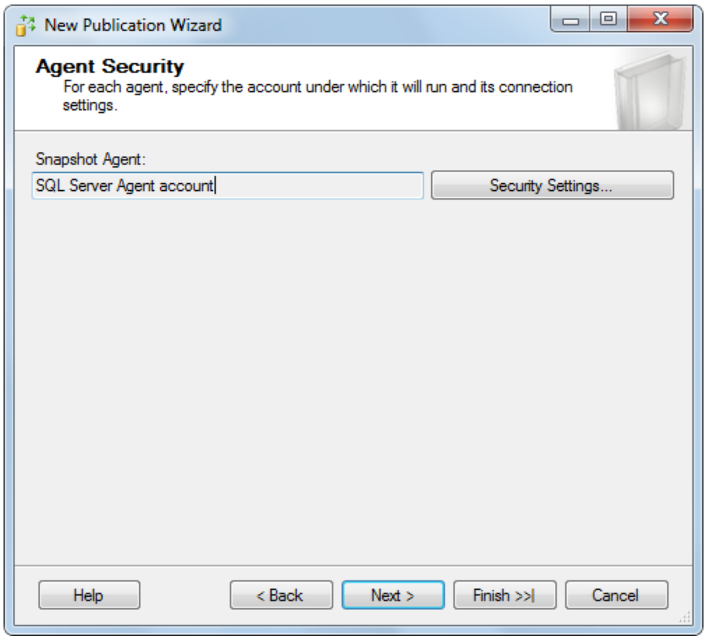
- **Create a snapshot immediately** - if enabled, the Snapshot Agent will take the snapshot immediately after this wizard is completed
- **Schedule the Snapshot Agent to run at the following times** - if enabled, the Snapshot Agent will take the snapshot on a regular basis, which can be specified if you click the *Change* button

Make the selection and click **Next**.



10. In this step, you need to configure the Snapshot Agent's security settings, i.e. how the Snapshot Agent will get authenticated when accessing the database to create the snapshot. The default **SQL Server Agent account** works fine in most cases, but you may need to specify different account in specific cases.

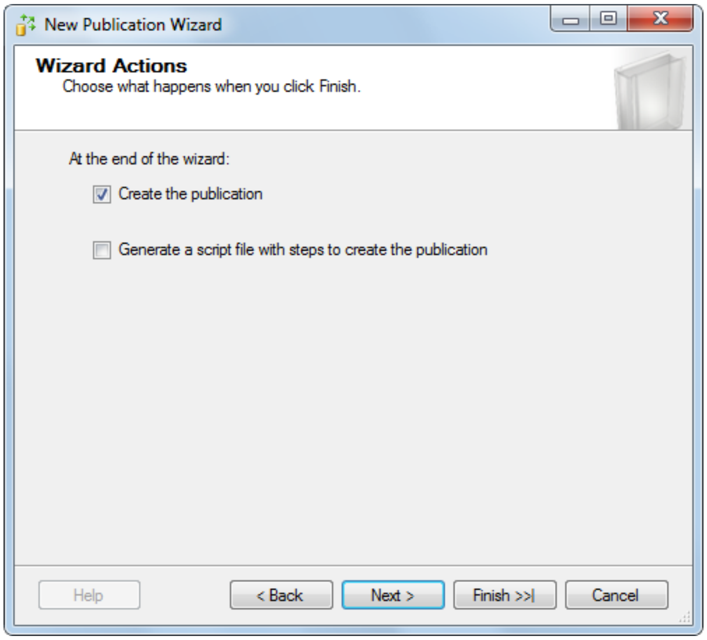
Make a configuration suitable for your environment and click **Next**.



11. The **Wizard Actions** step lets you decide what happens when you finish the wizard. You have the following two options:

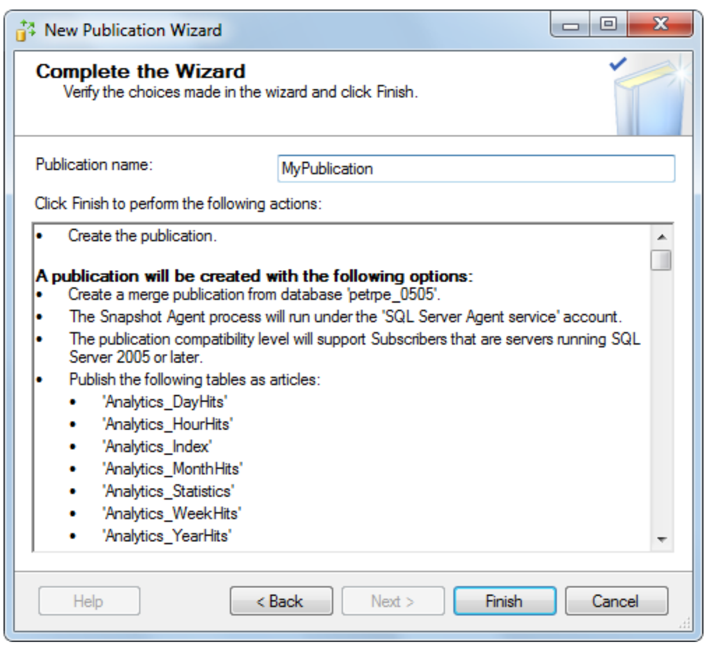
- **Create the publication** - if enabled, the publication will be created as defined throughout the wizard
- **Generate a script file with steps to create the publication -** if enabled, the wizard generates a script which, when executed, creates the publication as defined throughout the wizard

Click **Next**.



12. In the final step, you need to enter the name of your new publication into the **Publication name** field. The section below gives an overview of options which you defined throughout the wizard.

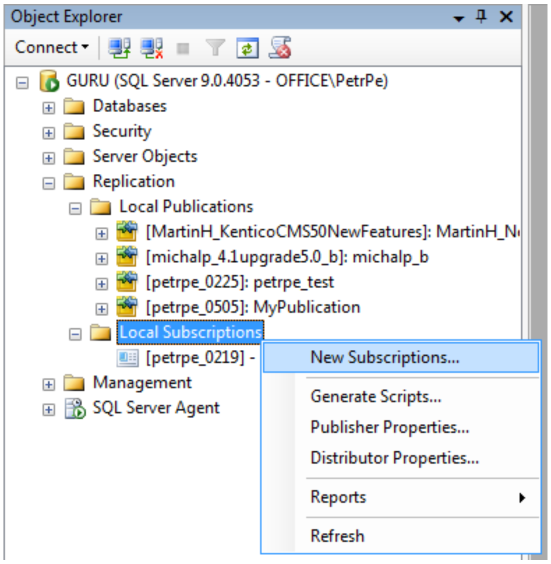
Click **Finish** to create the publication based on the listed options.



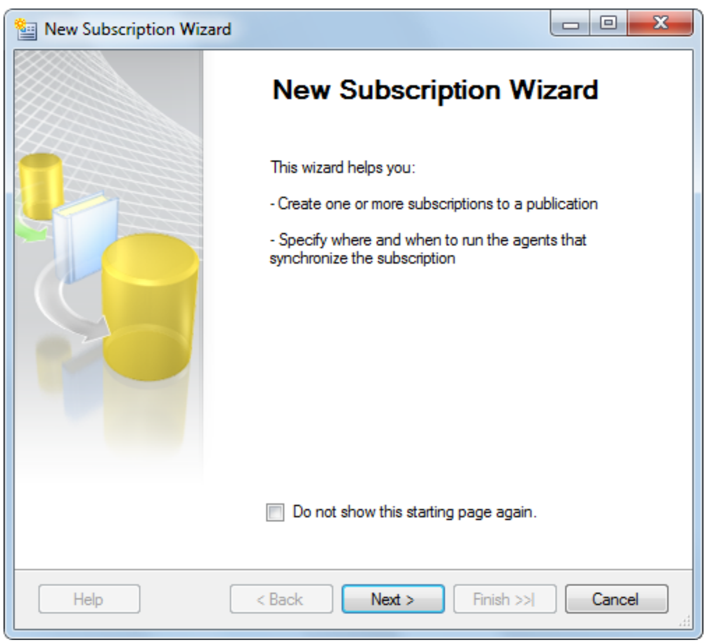
3.4.9.3 Creating a subscription

When you have a publication configured, you can proceed to creating the subscriptions. There can be one or more subscriptions, while you will need a dedicated database for each subscription. Each subscription database needs to be an **exact copy of the publication database**. The copy can be created using backup and restore in SQL Server Management Studio.

1. In **SQL Server Management Studio**, expand the **Replication** node and right-click the **Local Subscriptions** folder. Choose **New Subscriptions** from the context menu.

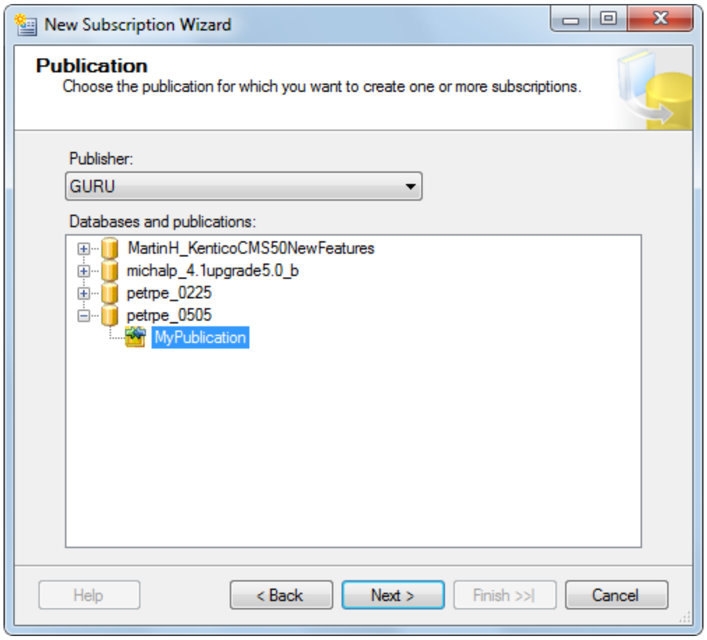


2. The **New Subscription Wizard** starts. Click **Next** in the first step.



3. In the **Publication** step, you need to select the publication to which you want to subscribe. Select the server where the publication is located from the **Publisher** drop-down list. All publications on the selected server will be listed in the **Databases and publications** section below.

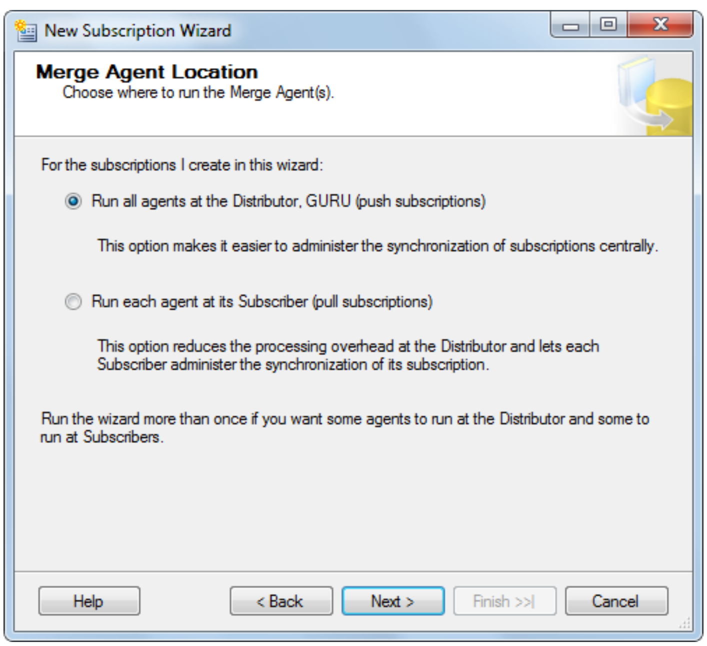
Select the required publication and click **Next**.



4. In this step, you need to decide where the Merge Agent will be run. You have the following two options:

- **Run all agents at the Distributor (push subscription)** - if selected, the Merge Agent will run at the distributor (the server where the publication is)
- **Run each agent at its Subscriber (pull subscription)** - if selected, the Merge Agent will run at the subscriber (the server where the subscription is)

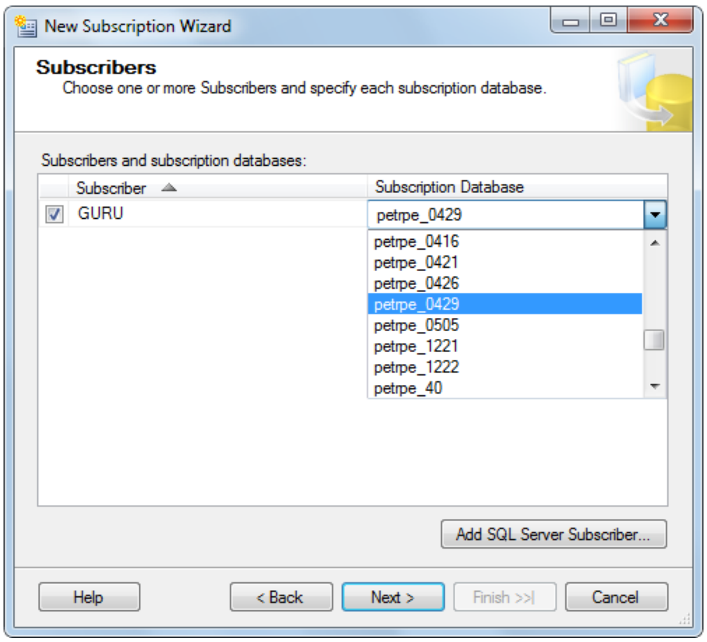
Make you choice (running the agent at the Distributor is recommended for most cases) and click **Next**.



5. Now you need to select the subscription database(s).

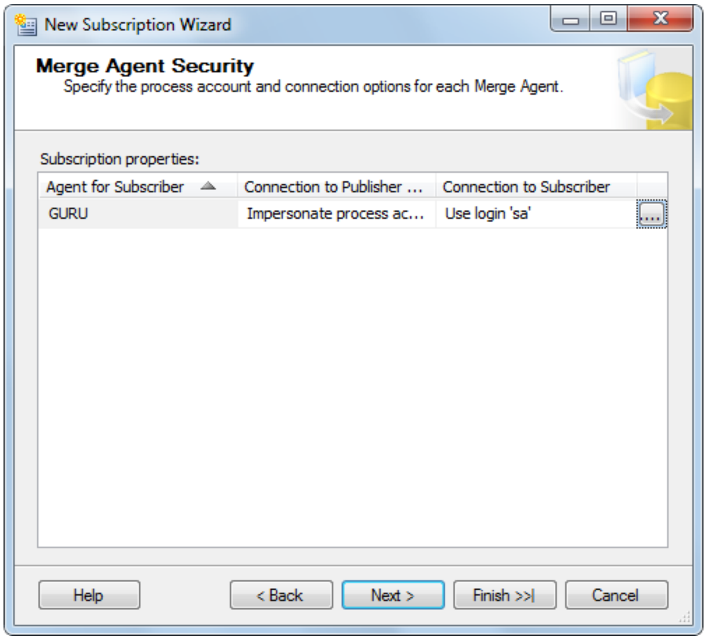
The currently managed database server will be offered in the **Subscriber** column. Using the **Add SQL Server Subscriber** button, you can add other servers to the list, which enables you to create subscriptions on a different server than the currently managed one. Next to each server, you can find a drop-down list containing all databases on the server.

Choose the subscription database(s) that you prepared before starting this wizard (as described at the beginning of this page) and click **Next**.



6. In the **Merge Agent Security** step, you are asked to specify the process account and connection options for the Merge Agent. This can be specified for each server by clicking the appropriate "**....**" button.

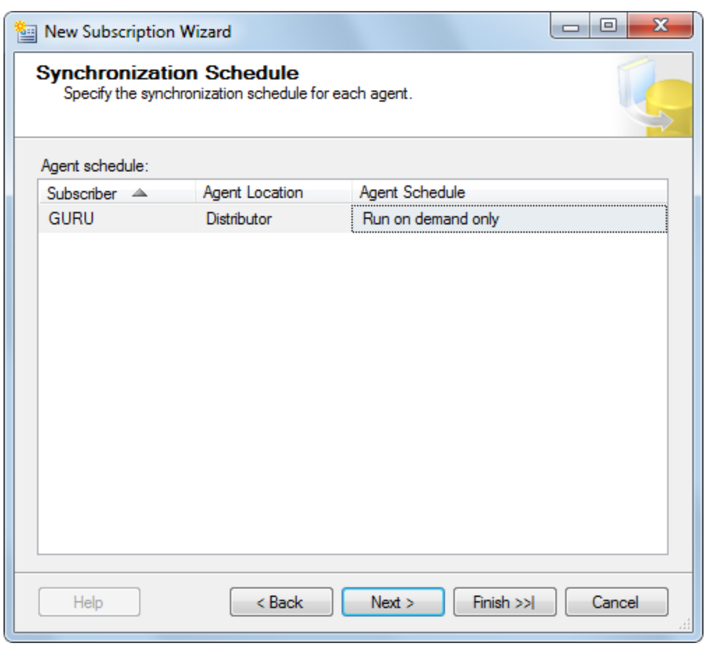
Specify the required information according to your environment and click **Next**.



7. In the **Synchronization Schedule** step, you need to specify when will synchronization between the subscriber and the publisher be performed. You have the following options:

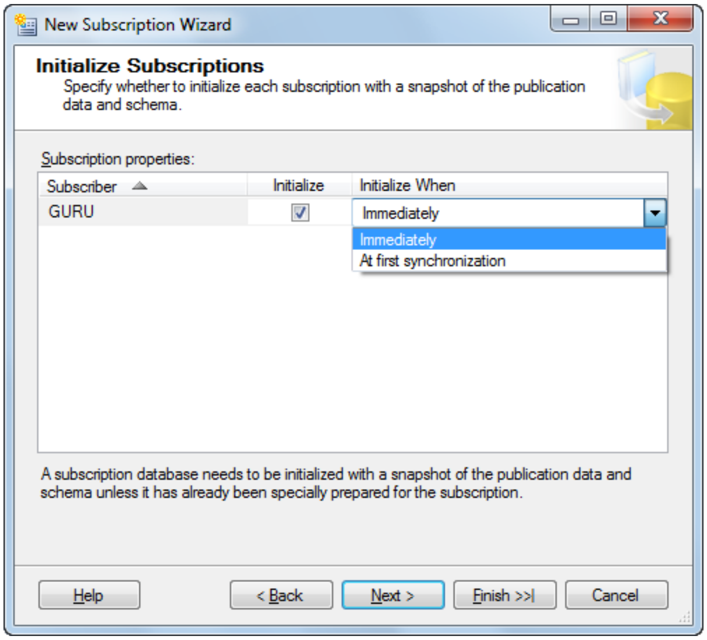
- **Run continuously** - synchronizes immediately whenever a change is made
- **Run on demand** - synchronization is performed only when executed manually from **Replication Monitor** in **SQL Server Management Studio**
- **Run on schedule** - synchronization is performed on a regular basis after a set interval

Select the required schedule and click **Next**.

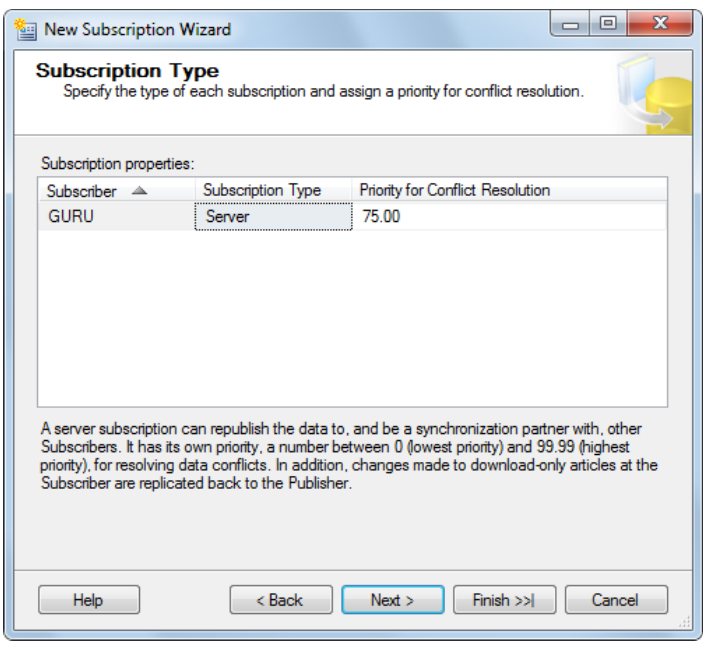


8. This step lets you initialize the subscription, i.e. fill the subscription database with data from the

publisher database snapshot. To perform this, the database snapshot needs to be already created. You can select from the following two options defining when will the synchronization be performed:



9. The **Subscription Type** step lets you decide if you want the subscription to be of the **Server** or **Client** type. For the purposes of our example, leave the default values (*Server* subscription with *75.00* priority) and click **Next**.

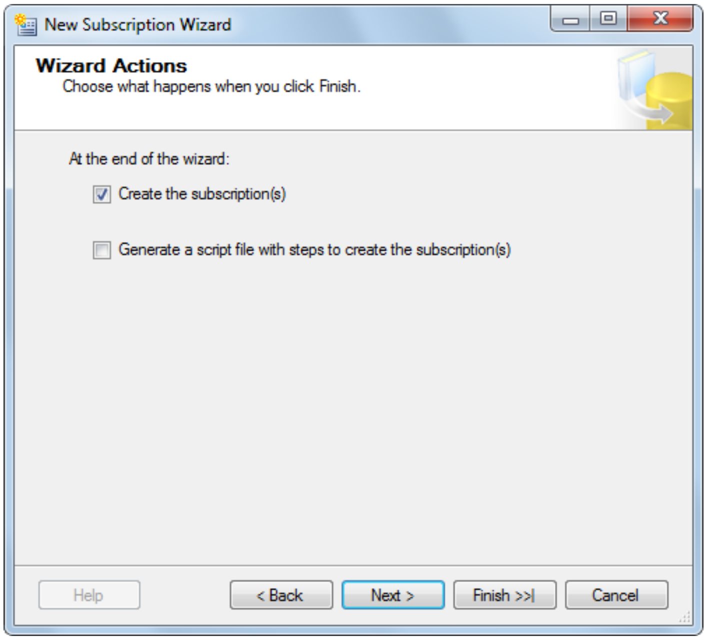


10. The **Wizard Actions** step lets you decide what happens when you finish the wizard. You have the following two options:

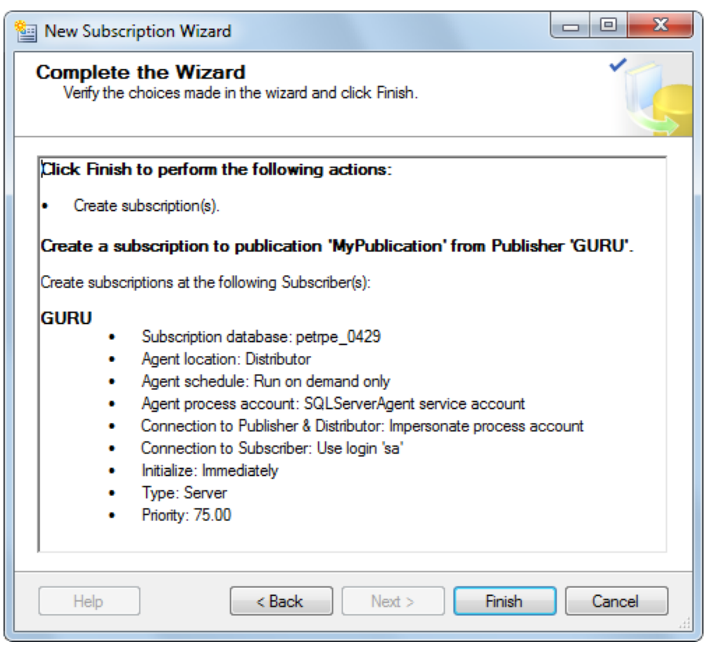
- **Create the subscription(s)** - if enabled, the subscription(s) will be created as defined throughout the

wizard
- **Generate a script file with steps to create the subscription(s)** - if enabled, the wizard generates a script which, when executed, creates the subscription(s) as defined throughout the wizard

Click **Next**.



11. The final step is only informational. It gives you an overview of the options that you selected throughout the wizard. Click **Finish** to create the publication based on the listed options.



3.4.9.4 Modifying structure of a replicated DB

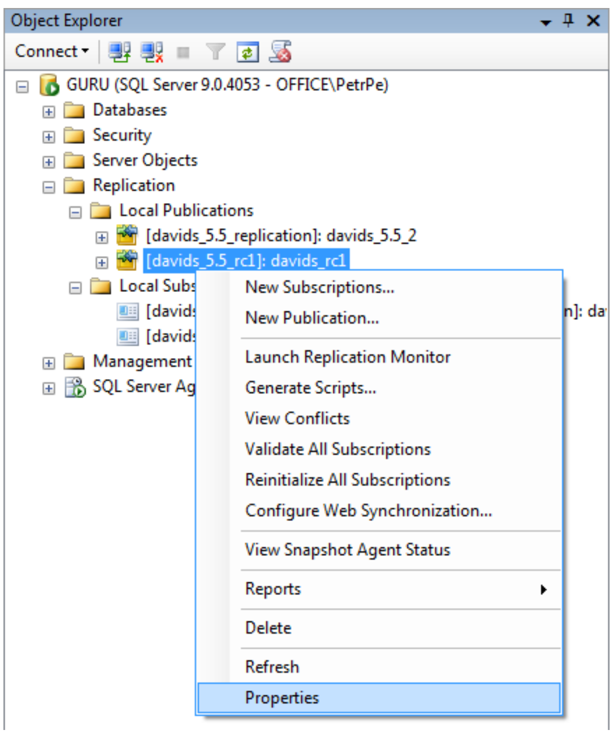
It is possible to make some changes to the DB structure even without disestablishing the existing replication.

Adding BizForms, Custom tables or Document types

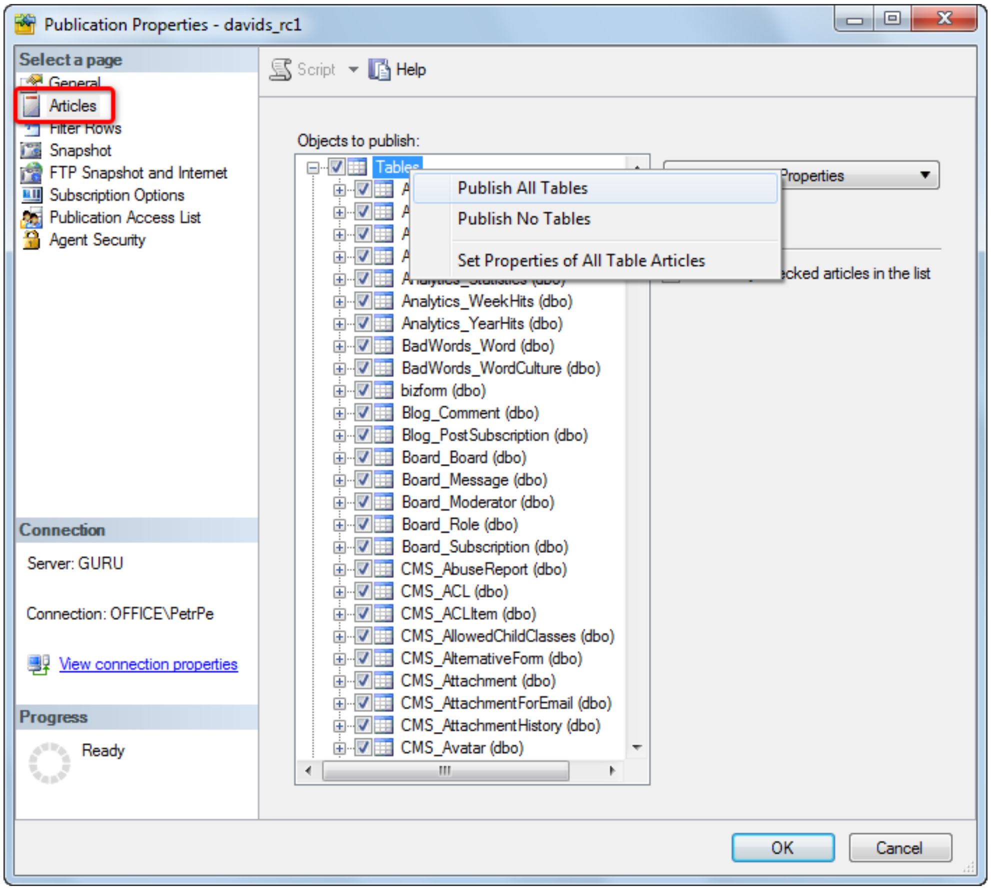
BizForms, Custom tables and Document types can be created even when database replication is established. You only need to follow the steps below to

1. Create the BizForm, Custom table or Document type from within the publisher's Kentico CMS user interface.

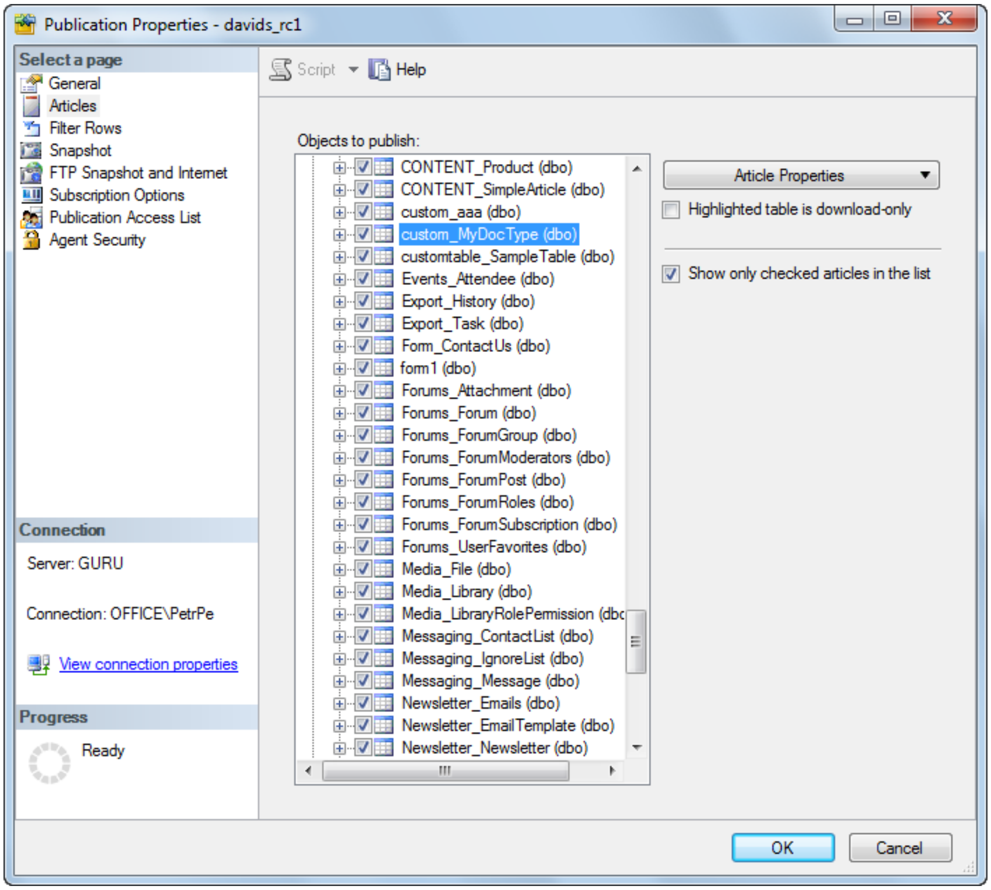
2. Open **SQL Server Management Studio**. In **Object Explorer**, expand the **Replication -> Local publications** node, right-click your publication and choose **Properties** from the context menu.



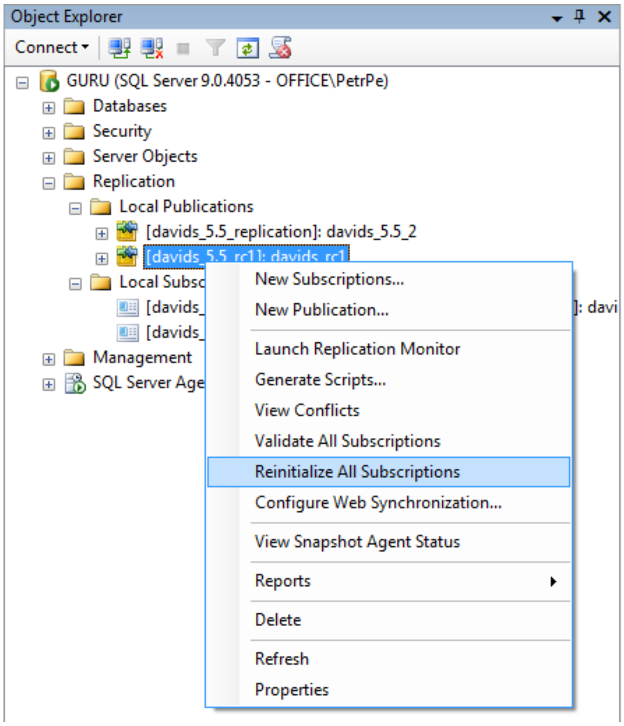
3. In **Publication Properties**, select **Articles** from the right menu. You will see a tree of database tables which are synchronized to the subscribers. You can verify that the new table created in step 1 is missing, i.e. it is not synchronized. Right-click **Tables** (the root of the tree) and click **Publish All Tables**.



4. In case that a new table is found, you are displayed with a confirmation window where you need to confirm that you really want to add the table (article) to the publication. Click **Yes**. The table will be added and you should be able to find it in the tree. Click **OK**.



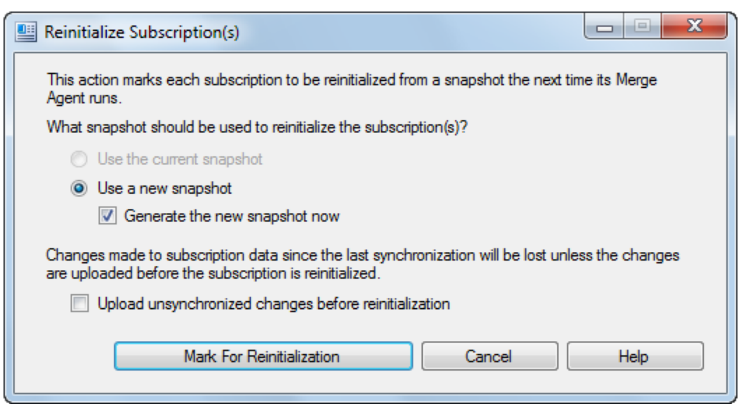
5. Now back in **Object Explorer**, right-click the publication again and choose **Reinitialize All Subscriptions** from the context menu.



6. A dialog window will be raised, telling you that all subscriptions need to be reinitialized from a snapshot the next time its Merge Agent runs. For the changes to be reflected, you need to reinitialize the subscriptions from a snapshot of the new database structure, i.e. from a snapshot which has been taken after steps 1-4 have been performed.

If you haven't created the snapshot since you finished step 4, choose **Use a new snapshot** and **Generate the new snapshot now**. If you already have the snapshot, choose **Use the current snapshot**.

Click **Mark For Reinitialization**. Next time the subscriptions' Merge Agent runs, the subscriptions will be reinitialized with the new database structure.



Field changes in System tables

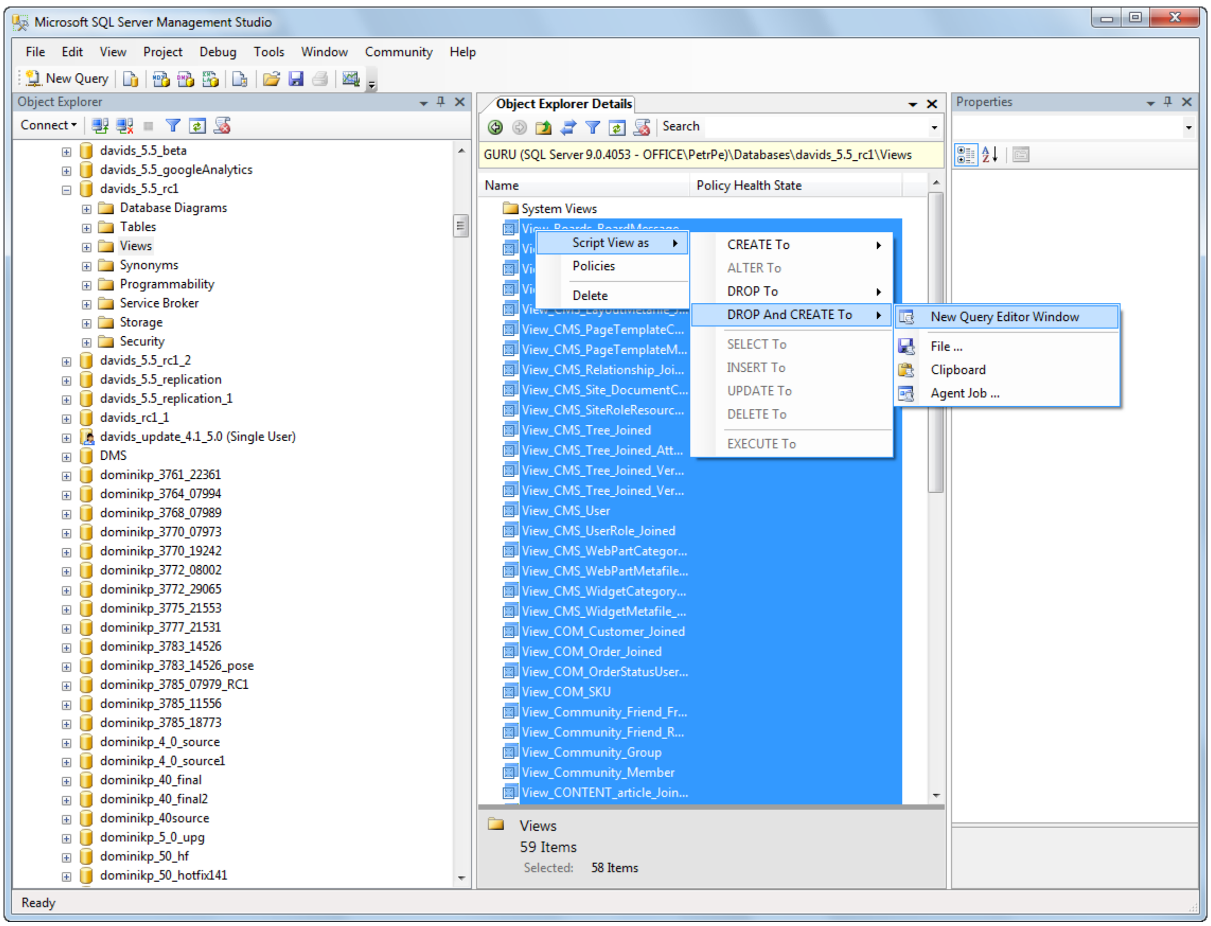
It is also possible to make changes to the fields of System tables without disestablishing an existing replication. In this case, you need to replace all views on the subscription databases with views from the modified publisher database.

The easiest way to do this is to generate a DROP And CREATE script on the publication database and run it on the subscription databases.

1. Click **F7** to open **Object Explorer Details** view and select the **Views** folder of the publication database.

2. Select all listed views, right-click and select **Script View as -> DROP And CREATE To -> New Query Editor Window** from the context menu.

3. Execute the generated script on the subscription databases.



Field changes in Document types

Changes to document type fields can also be made without disestablishing an existing replication. The solution in this case is almost identical to the solution for system tables (described above). The only difference is that you don't need to DROP And CREATE all views, but only the view related to the particular document type.

3.4.10 Configuration of full-text search in files

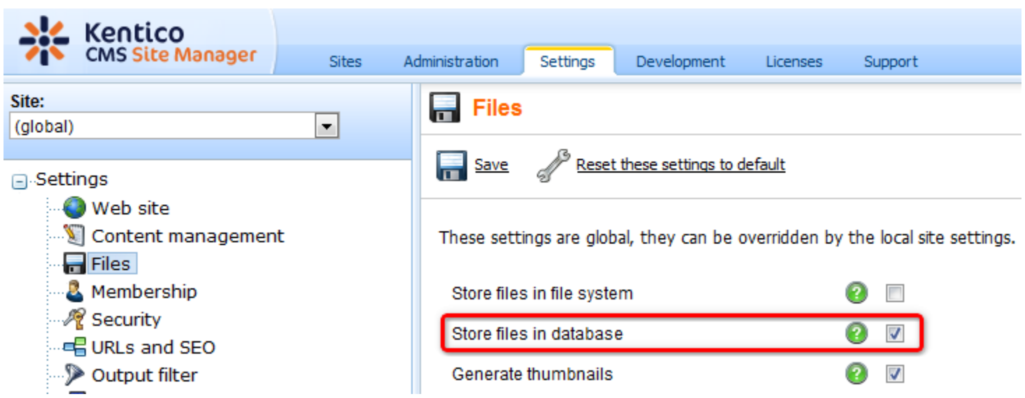
3.4.10.1 Overview

Kentico CMS allows you to full-text search files uploaded into the database. It uses standard Microsoft SQL Server Search Engine that allows you to index a wide variety of files. This chapter explains how you can configure your installation of Kentico CMS for search in files.

Prerequisites - Important!

If you're using Microsoft SQL Server 2005, you need to ensure that full-text search support is installed on the server. The full-text search is available for all editions of Microsoft SQL Server 2005, including Express Edition with Advanced Services.

Generally, full-text search support must be installed on your SQL Server and your website must be configured for storing files in the database (**Site Manager -> Settings -> Files -> Store files in database** must be enabled).



Now follow one of the following guides:

- Configuration of full-text search on Microsoft SQL Server 2005
- Configuration of full-text search on Microsoft SQL Server 2005 Express Edition

Supported file types

The standard full-text search engine delivered with Microsoft SQL Server can search TXT, HTML, DOC, XLS and PPT files.

You can install a free driver from Adobe for searching in PDF files as described in the Searching PDF files topic.

To search Microsoft Office 2007 and 2010 document files such as DOCX or XLSX, it is necessary to install a free IFilter pack as described in the Searching Office 2007/2010 documents topic.

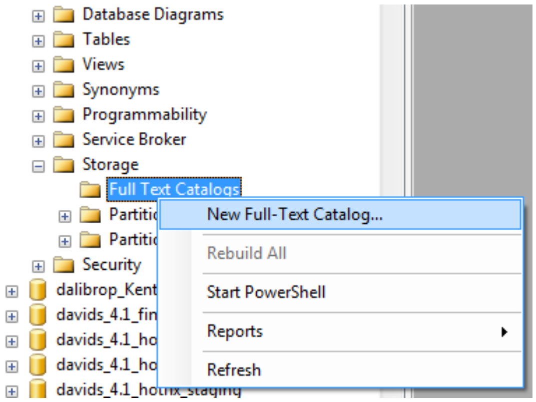
If you want to search other types of documents, you need an appropriate IFilter library. IFilter libraries can be purchased from third-party vendors (you can find them using

Google).

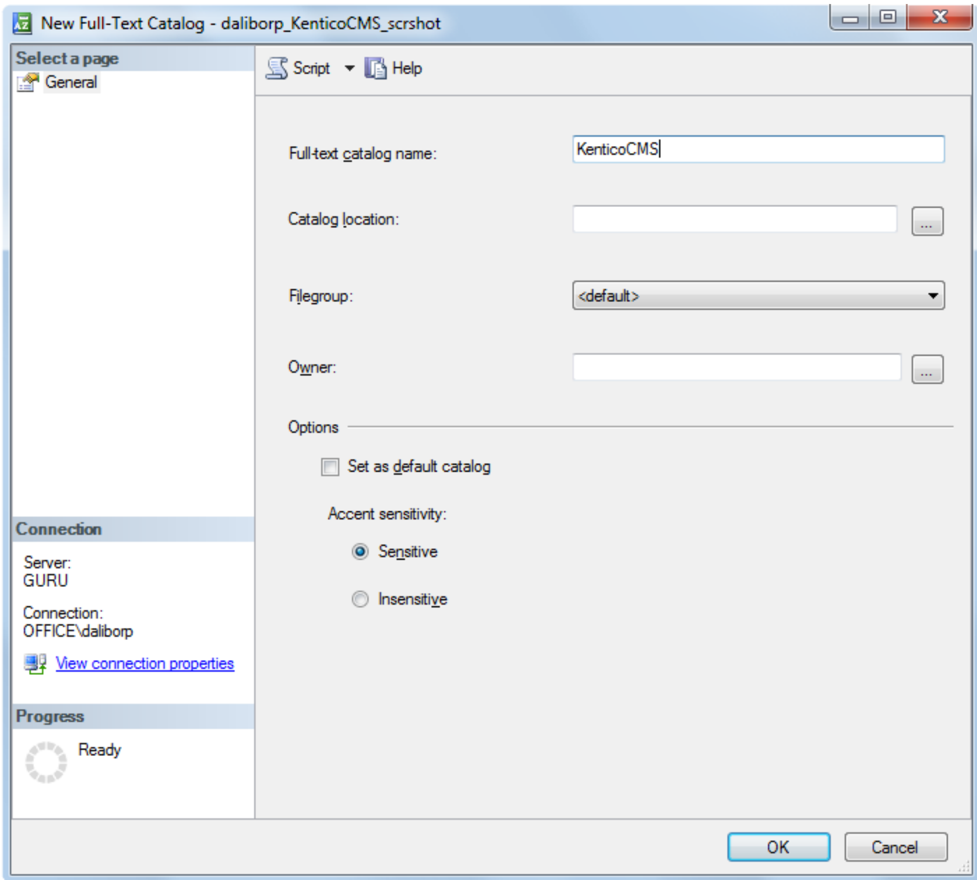
3.4.10.2 Configuration on MSSQL 2005 and 2008

1. Start **Microsoft SQL Server Management Studio**.

2. Locate your database with Kentico CMS objects, unfold **Storage**, right-click **Full Text Catalogs** and choose **New Full-Text Catalog**.

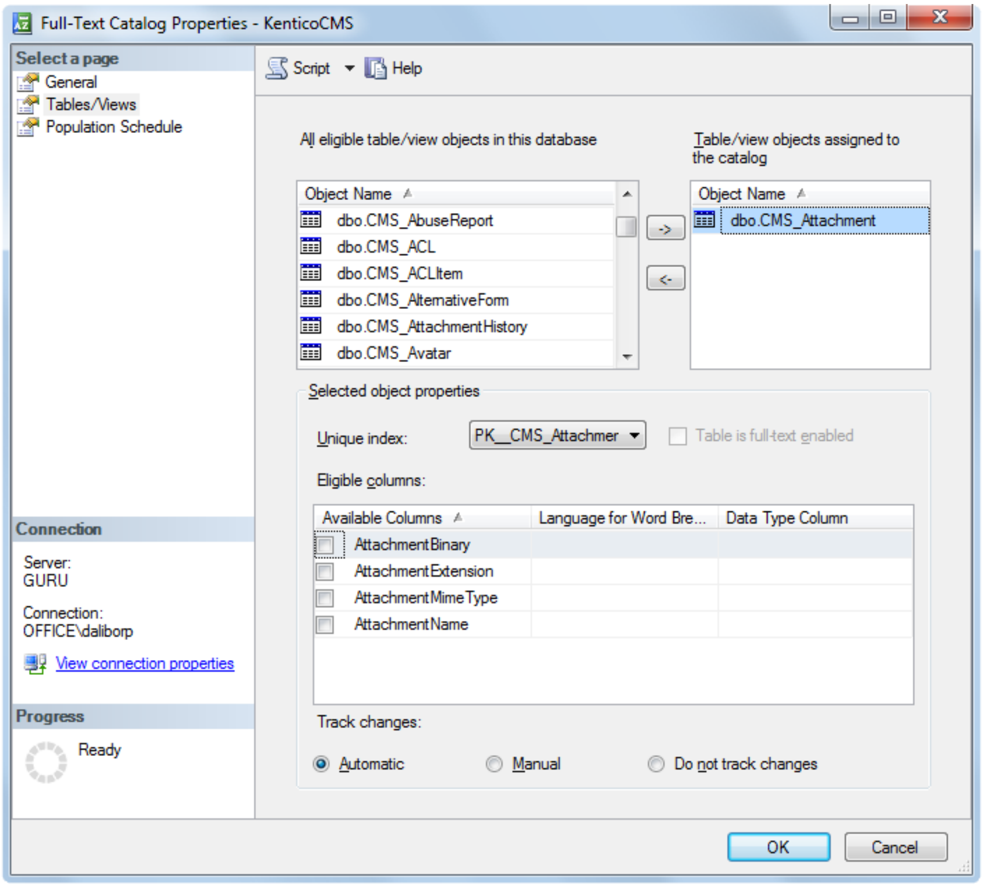


3. On the **New Full-Text Catalog** dialog, enter Full-text catalog name *KenticoCMS* and click **OK**.



4. Right-click the newly created KenticoCMS full-text catalog and choose **Properties**. On the Full-Text Catalog Properties dialog, choose the **Tables/Views** tab.

5. Assign the *CMS_Attachment* table to the catalog, check the box next to the *AttachmentBinary* column, set the **Language for Word Breaker** to *English* or other value and set the **Data Type Column** to *AttachmentExtension* as shown on the following figure and click **OK**.



6. Sign in as administrator and go to **Site Manager -> Development -> Document types -> Root -> Queries** and edit the **searchattachments** query. Uncomment the following query:

```
SELECT view_cms_tree_joined.*, view_cms_tree_joined.NodeName AS SearchResultName
FROM cms_attachment
INNER JOIN view_cms_tree_joined on view_cms_tree_joined.DocumentID =
cms_attachment.AttachmentDocumentID
WHERE NodeAliasPath LIKE @NodeAliasPath
AND
([AttachmentName] Like N'%'+ @Expression + N'%' OR FREETEXT(attachmentbinary,
@expression))
AND @SiteName = SiteName AND (@DocumentCulture = DocumentCulture OR
@DefaultCulture = DocumentCulture )
```

Note: It is not necessary to uncomment the query if a **Smart search** web part is used for search. In that case the **Search in attachments** option in **Smart search dialog with results** has to be checked.

3.4.10.3 Configuration on MSSQL 2005 Express Edition

If you cannot use the management console for full-text search configuration, you can use the following SQL script:

```
-- Allow IFilter library loading
exec sp_fulltext_service 'verify_signature', 0
exec sp_fulltext_service 'load_os_resources', 1

-- Create Full-Text catalog
exec sp_fulltext_catalog 'MyKenticoCMSCatalog','create'

-- Add Full-Text catalog to table
exec sp_fulltext_table 'CMS_Attachment','create','MyKenticoCMSCatalog',
'PK_CMS_Attachment'

-- Add Index to table column
exec sp_fulltext_column 'CMS_Attachment','AttachmentBinary','add',NULL,
'AttachmentExtension'

-- Populate the catalog
exec sp_fulltext_table 'CMS_Attachment','start_full'
```

Please note: Similarly to Configuration on MSSLQ 2005 and 2008, you need to go to **Site Manager -> Development -> Document types -> Root -> Queries**, edit the **searchattachments** query and uncomment the following query to be able to run full-text search.

```
SELECT view_cms_tree_joined.*, view_cms_tree_joined.NodeName AS SearchResultName
FROM cms_attachment
INNER JOIN view_cms_tree_joined on view_cms_tree_joined.DocumentID =
cms_attachment.AttachmentDocumentID
WHERE NodeAliasPath LIKE @NodeAliasPath
AND
([AttachmentName] Like N'%'+ @Expression + N'%' OR FREETEXT(attachmentbinary,
@expression))
AND @SiteName = SiteName AND (@DocumentCulture = DocumentCulture OR
@DefaultCulture = DocumentCulture )
```

Note: It is not necessary to uncomment the query if a **Smart search** web part is used for search. In that case the **Search in attachments** option in **Smart search dialog with results** has to be checked.

3.4.10.4 Searching PDF files

Searching in PDF files is not supported in Kentico CMS by default. To enable it, you need to install the free **Adobe PDF IFilter** on your machine. Please note that Adobe PDF IFilter is not a product of Kentico Software and we cannot guarantee its functionality.

The installation differs for 32-bit and 64-bit systems. Both are described in the text below.

Configuration on 32-bit platforms

The following procedure needs to be followed to enable searching in PDF files on 32-bit platforms:

1. Download the free Adobe PDF IFilter 6.0 from the Adobe website: http://www.adobe.com/support/downloads/detail.jsp?ftpID=2611

2. Run the installer and follow the instructions.

3. After you finish the installation, rebuild the full-text search catalog that you created for Kentico CMS. Now you should be able to search PDF files.

4. If you're using MS SQL Server 2005, you may need to run the following SQL commands so that the server can load PDF IFilter libraries:

```
sp_fulltext_service 'verify_signature', 0
GO
sp_fulltext_service 'load_os_resources', 1
```

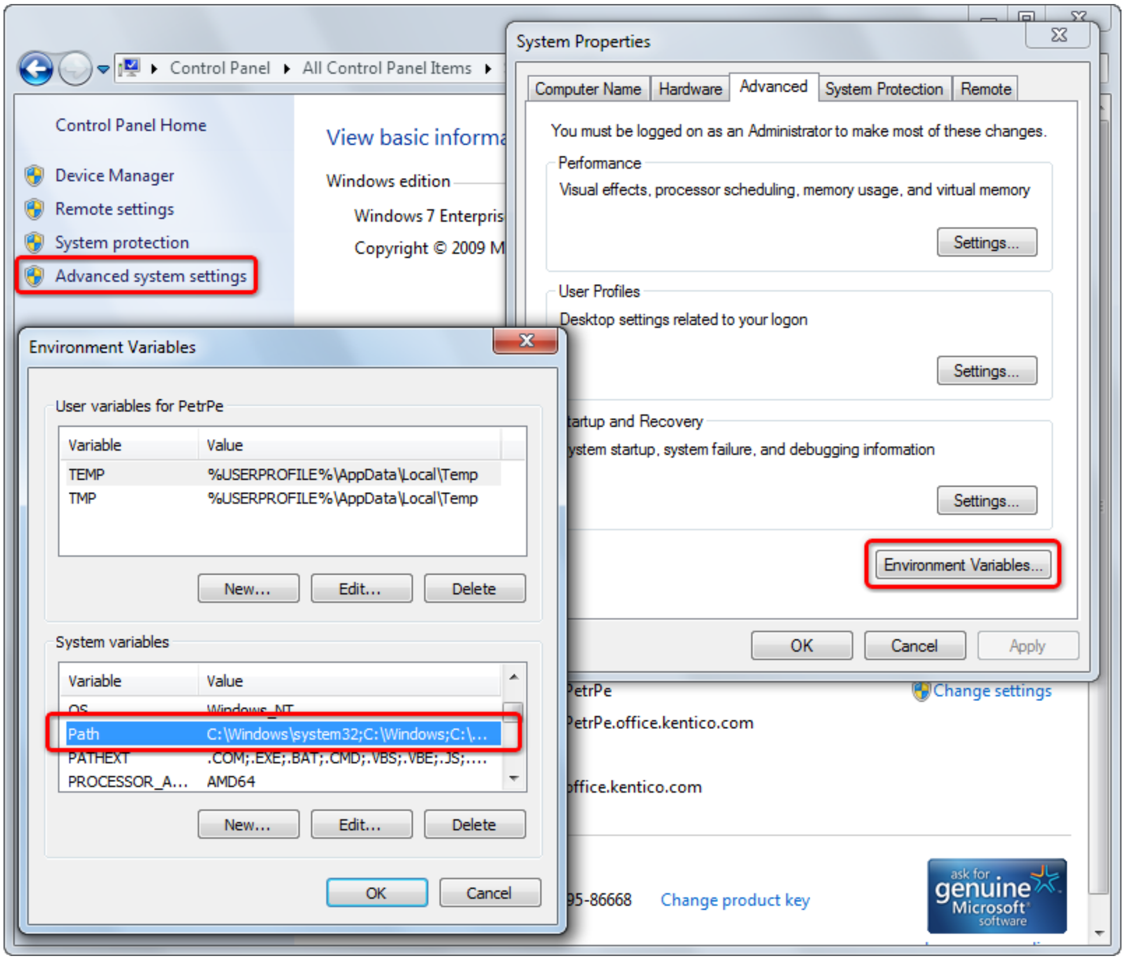
Configuration on 64-bit platforms

The following procedure needs to be followed to enable searching in PDF files on 64-bit platforms:

1. Download Adobe PDF IFilter 9 for 64-bit platforms from the following location: http://www.adobe.com/support/downloads/detail.jsp?ftpID=4025

2. Make sure that no other version of IFilter is installed on your machine, run the installer and follow the instructions.

3. After installing the PDF IFilter, it is recommended to set your system **Path** environment variable to the **bin** folder of the IFilter installation. Go to **Control Panel -> System -> Advanced system settings -> Environment Variables** and append *C:\Program Files\Adobe\Adobe PDF IFilter 9 for 64-bit platforms\bin* to the value of the *Path* variable.



4. Restart the computer. When restarted, check whether Adobe IFilter is registered with the server - in a new query window, type and execute the following:

```
SELECT * from sys.fulltext_document_types
```

You should see a list of installed filters in the output window. Verify if you can see an entry for **.pdf** with a correct path set.

5. If you do not see an entry for **.pdf**, you need to execute the following two lines:

```
EXEC sp_fulltext_service 'load_os_resources', 1
EXEC sp_fulltext_service 'verify_signature', 0
GO
```

6. Restart the SQL Server. Run the query from step 4 above again and verify that you can see the entry for **.pdf**.

3.4.10.5 Searching Office 2007/2010 documents

Searching in Microsoft Office 2007/2010 documents is not supported in Kentico CMS by default. To enable it, you need to install and configure **Microsoft Filter Pack** IFilters on your machine. Please note that Microsoft Filter Pack is not a product of Kentico Software and we cannot guarantee its functionality.

The following procedure needs to be completed to enable searching in the desired file types:

1. Download the Microsoft Filter Pack installer that matches the architecture of your operating system from the Microsoft website:

- Microsoft Office 2007 Filter Pack
- Microsoft Office 2010 Filter Pack

2. Run the installer and follow the instructions.

3. After you finish the installation, start SQL Server Management Studio, select the instance that contains your Kentico CMS database and run the following command in that instance:

```
sp_fulltext_service 'load_os_resources', 1
```

This allows the server to load the IFilter libraries.

4. Check whether the IFilters are registered with the server. Type and execute the following in a new query window:

```
SELECT * from sys.fulltext_document_types
```

You should see a list of installed filters on the **Results** tab of the output window. Verify that the list contains entries for all desired file extensions with a correct path set.

5. Restart the SQL Server service. Once this is done, **Rebuild** the full-text search catalog that you created for Kentico CMS. Now you should be able to search Office 2007/2010 files.

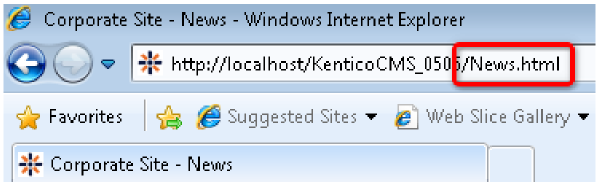
3.4.11 Custom URL extensions and extensionless URLs

3.4.11.1 Overview

This chapter describes how to configure the system for using custom URL extensions and extension-less URLs. Custom URL extensions let you have the extensions rendered with a different extension than the default *.aspx*, like *.html*, *.php*, *.xxx* and any other.

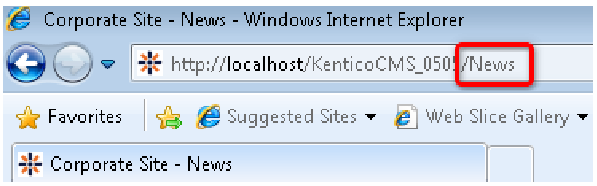
So for example, instead of the default *http://www.example.com/news**.aspx***, you can have URLs ending with **.html**:

*http://www.example.com/news**.html***



you can also configure an empty extension, which results in an **extension-less URL**:

http://www.example.com/news



To achieve this result you have to take the following two steps:

1. Adjust your web.config or IIS to handle the custom extensions. This step is different for IIS 7 or higher and IIS 6.
2. Configure custom URL extensions in Kentico CMS. It is described in this chapter.

3.4.11.2 IIS 7 and higher

If your instance of Kentico CMS is running on a server with IIS 7 or higher installed, the procedure is quite simple - it consists only of adding and modifying a few keys in your project's **web.config** file.

For it to work, you need to modify the *system.webServer* section of your **web.config** file by adding the first highlighted line and changing the opening <*modules*> tag to the second highlighted line in the code example below.

```
<system.webServer>
    ...
 <httpErrors existingResponse="PassThrough" />
 <modules runAllManagedModulesForAllRequests="true">
    ...
</system.webServer>
```

Please be aware that this procedure only works for IIS 7 sites that use an **Application Pool** with **Managed Pipeline Mode** set to *Integrated*. This should be the default setting.

With these modifications in your **web.config**, you can proceed to configuring the extensions in Kentico CMS administration interface. You can also perform some additional configuration as described below.

It is recommended to configure a custom *page-not-found* error page when using custom URL extensions, otherwise a blank page may be displayed in some browsers if a user attempts to access a non-existing resource. This can be achieved by specifying the appropriate page as the value of the **Page not found URL** setting at **Site Manager -> Settings -> Web site -> Web site content**, for example: *~/CMSMessages/PageNotFound.aspx*.

You can also add the following key to the *AppSettings* section of your **web.config** file, which ensures that URLs remain the same even after postback.

```
<add key="CMSUseExtensionOnPostback" value="false" />
```

If you are using trailing slashes (enabled by the **Use URLs with trailing slash** option in **Site Manager -> Settings -> URLs and SEO**), you can use the following extra key to have only extensionless URLs ending with the trailing slash. URLs ending with an extension are rendered without the slash when the key is used.

```
<add key="CMSUseTrailingSlashOnlyForExtensionLess" value="true" />
```

Using the cmspages/handler404.aspx special page (obsolete)

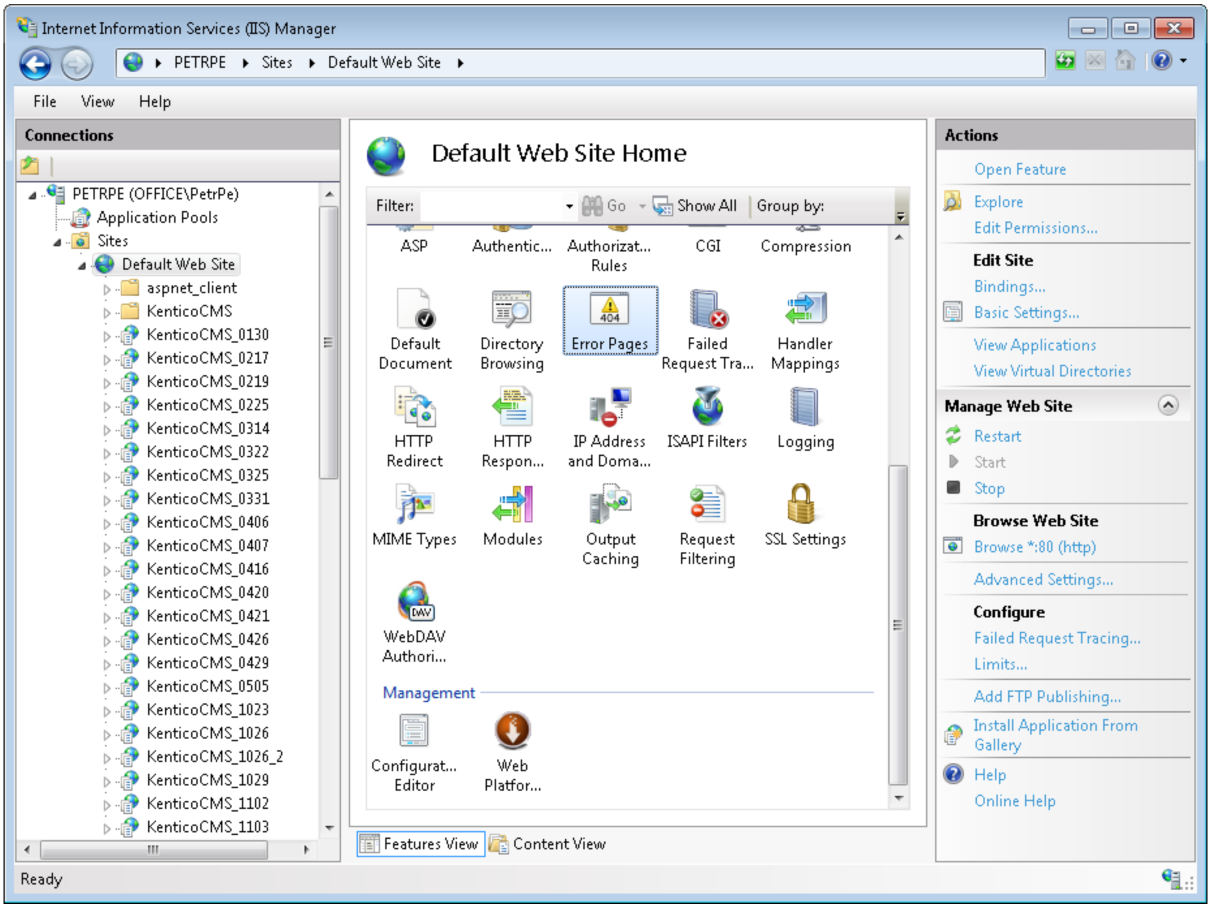
Due to backward compatibility, you can also enable custom URLs by configuring your IIS manually. It is not recommended now as it is obsolete.

If you receive the **Lock violation error** during the procedure, try the solution described here.

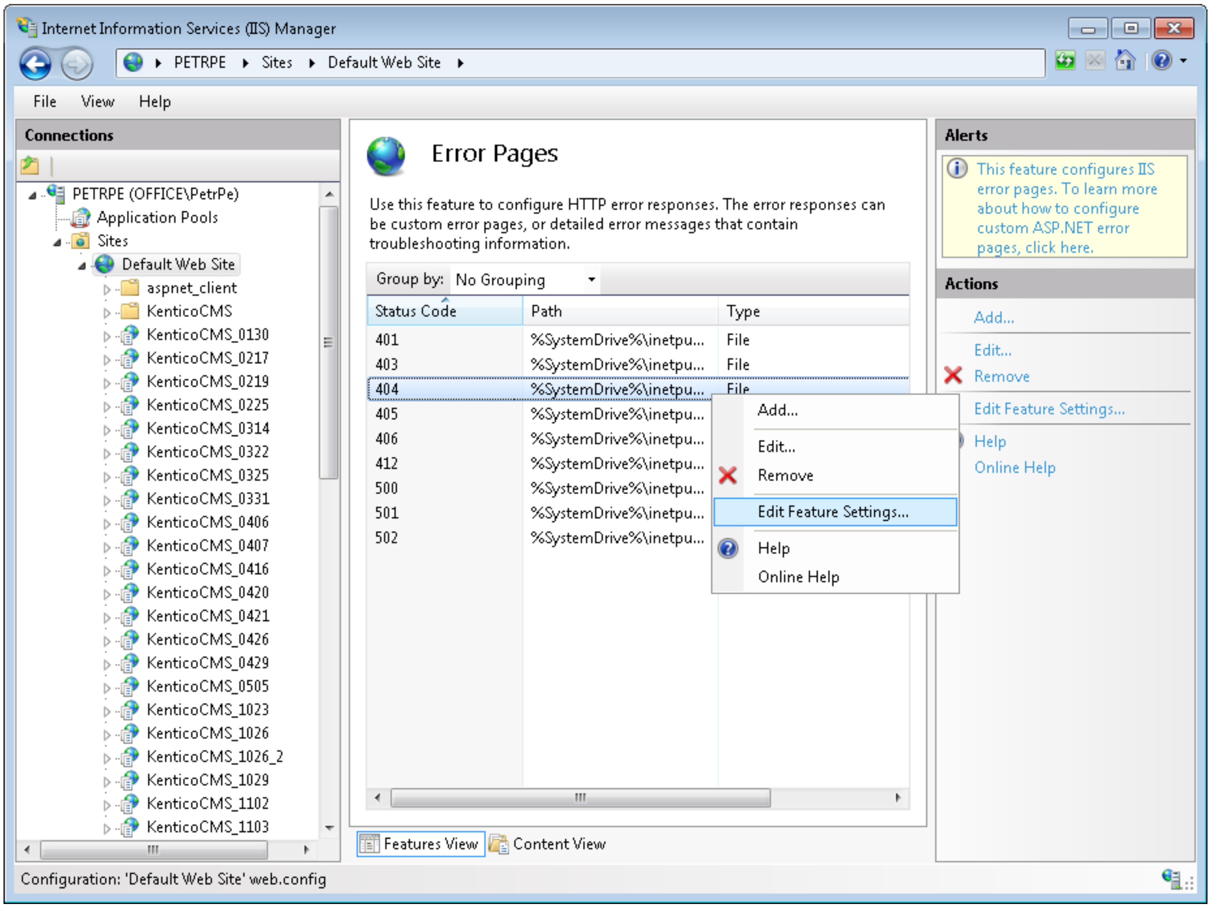
The setup procedure on IIS 7 is the following:

1. Open **Start -> Control Panel -> Administrative Tools -> Internet Information Services (IIS) Manager**.

2. Select your website from the tree on the left and open the **Error pages** section.



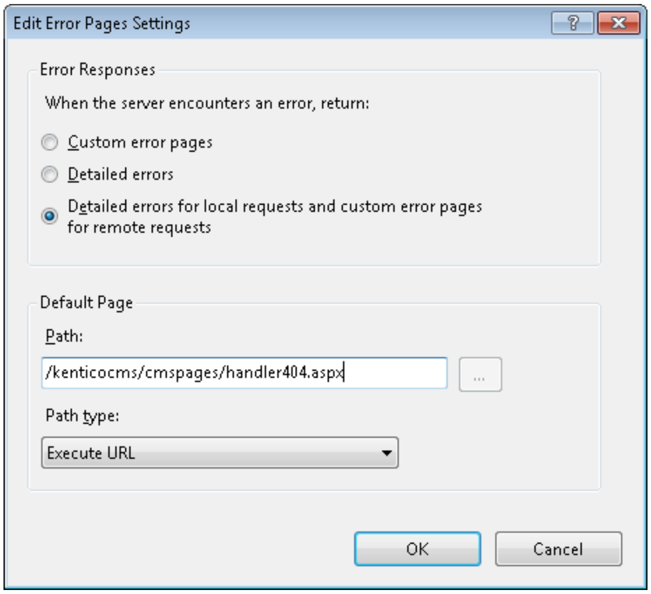
3. Right-click the **404 error** and choose **Edit Feature Settings**.



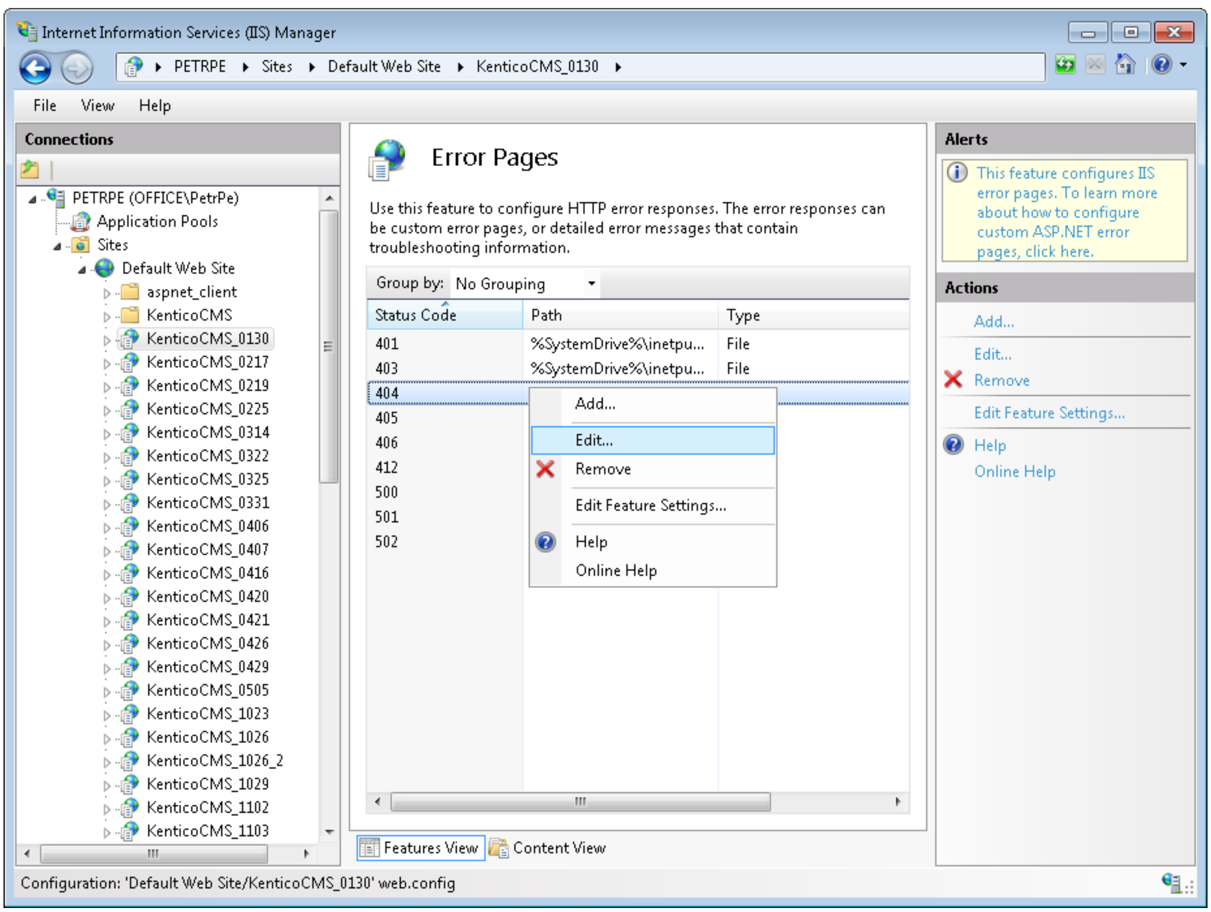
4. Enter the following values:

- **Path:** enter the URL of the **cmspages/handler404.aspx** page according to your application's URL.
  Example: if you run your web project in virtual directory `/kenticocms`, you need to enter `/kenticocms/cmspages/handler404.aspx`
- **Path type:** Execute URL

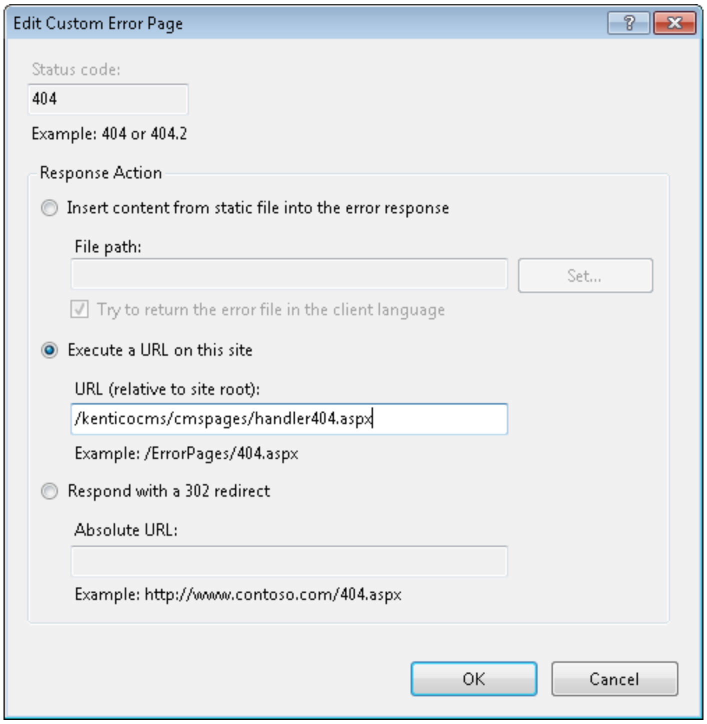
Click **OK**.



5. Now right-click the **404 error** again and choose **Edit** from the context menu.



6. Select **Execute a URL on this site** and enter the same URL that you entered in step 4. Click **OK**.



7. Go back to step 3 and repeat the same procedure for the **405 error**.

8. Click **OK** on all dialogs to save the changes. It's not necessary to restart the application.

Setting up the .html extension on IIS manually (obsolete)

The following approach is also possible, but not recommended and considered obsolete now. To use the **.html** extension, go through the following steps:

1. Run **IIS Manager** .

2. Select your web.

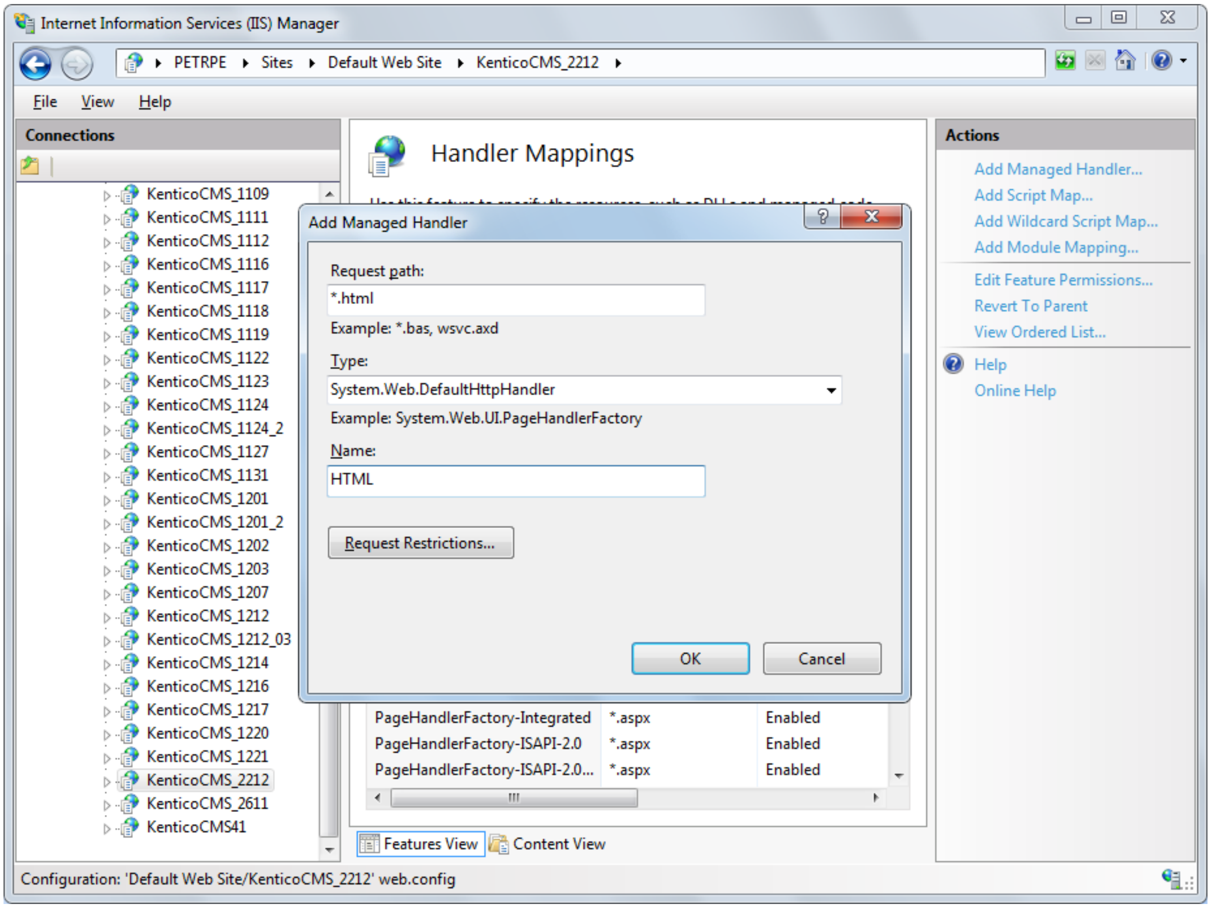
3. Open **Handler mappings**.

4. Click to **Add managed handler...** .

5. Enter the following values:

- **Request path**: *.html
- **Type**: System.Web.DefaultHttpHandler
- **Name**: HTML

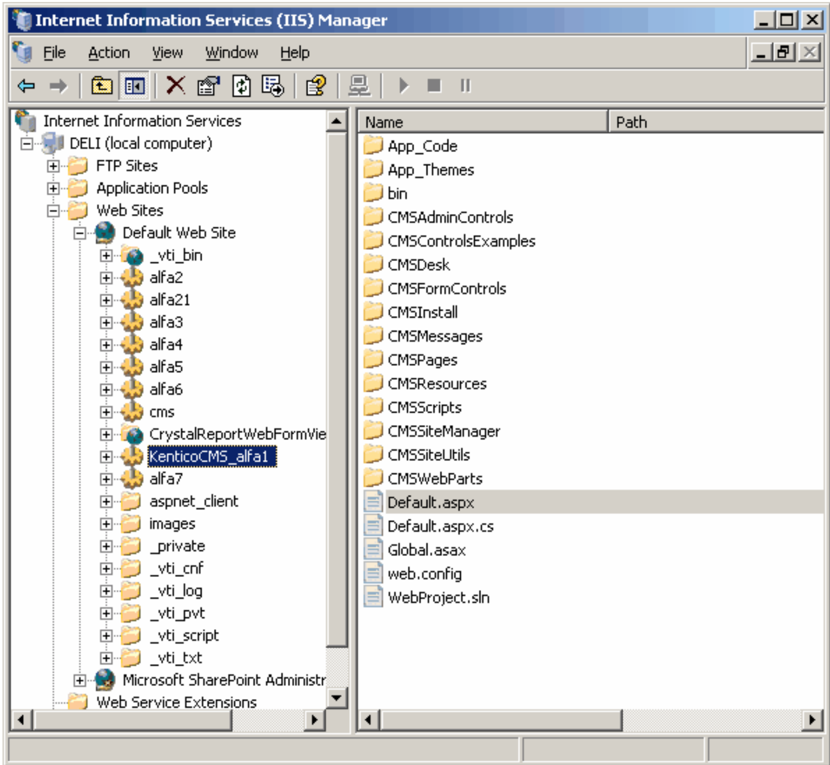
Click **OK**.



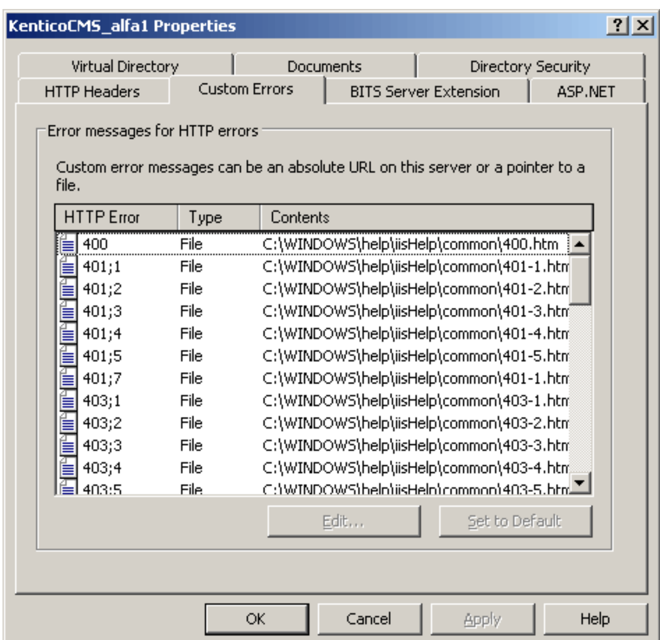
3.4.11.3 IIS 6

This procedure needs to be performed to configure IIS 6 to hande Kentico CMS's custom URL extensions. You can use this configuration on **Windows XP** and **Windows Server 2003** with Internet **Information Services (IIS) 6** installed. It's not possible to use it with Visual Studio's built-in web server.

1. Go to **Start -> Control Panel -> Administrative Tools** and launch the **Internet Information Services (IIS) Manager**. Locate the appropriate website and virtual directory (if you installed Kentico CMS into the root, you will make this change on the website level only).



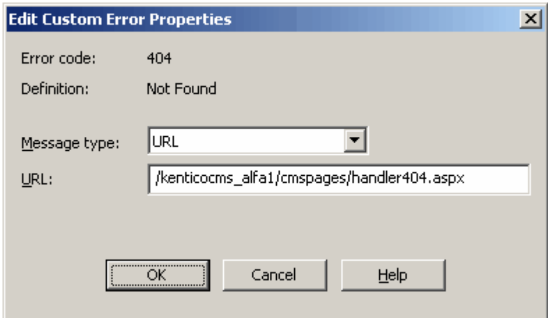
2. Right-click the directory (or website) and choose **Properties** and then click the **Custom Errors** tab:



3. Edit the 404 error and enter the following values:

- **Message type:** URL
- **URL:** enter the URL of the cmspages/handler404.aspx page according to your application's URL. E. g. if you run your web project in virtual directory `/kenticocms`, you need to enter `/kenticocms/`

```
cmspages/handler404.aspx
```



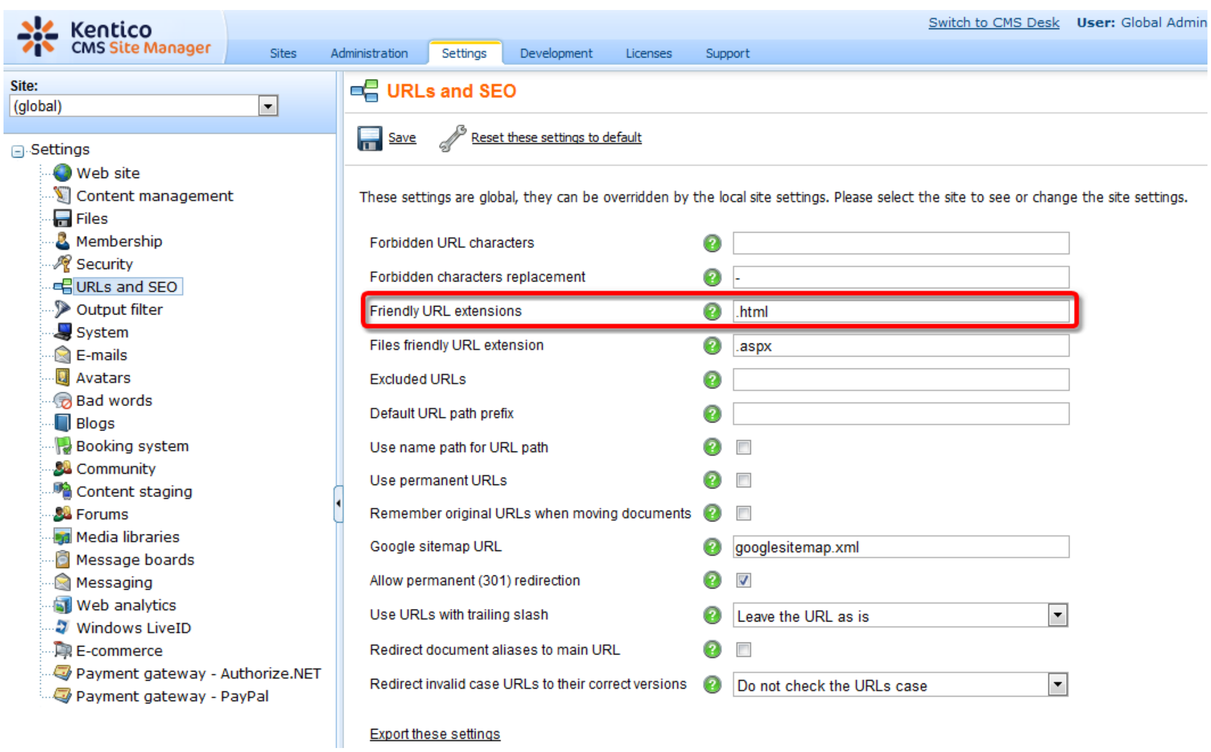
4. Now repeat the same for the 405 error, using the same custom URL: `/kenticocms/cmspages/handler404.aspx`

Click **OK** on all dialogs to save the changes. It's not necessary to restart the application.

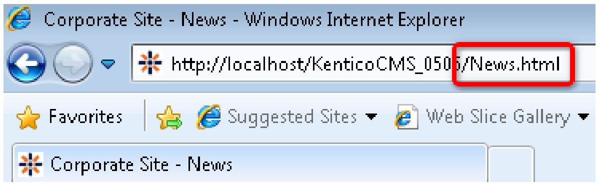
3.4.11.4 Configuration of custom URL extensions (.html or other)

When you have performed the required configuration for your version of IIS, you can proceed with entering the actual extensions.

This can be done in **Site Manager -> Settings -> URLs and SEO**. For example, try to set the **Friendly URL extensions** value to *.html* and click **Save**.



Now when you go to the live website, you will see that all URLs in menus and listings are rendered with **.html** extensions. In some cases, you may need to update some static links which were created with the default **.aspx** extension.



Using multiple extensions

You can enter multiple extensions into the **Friendly URL Extension** field mentioned above. The following format should be used:

.html;.htm;;.xxx;.abc

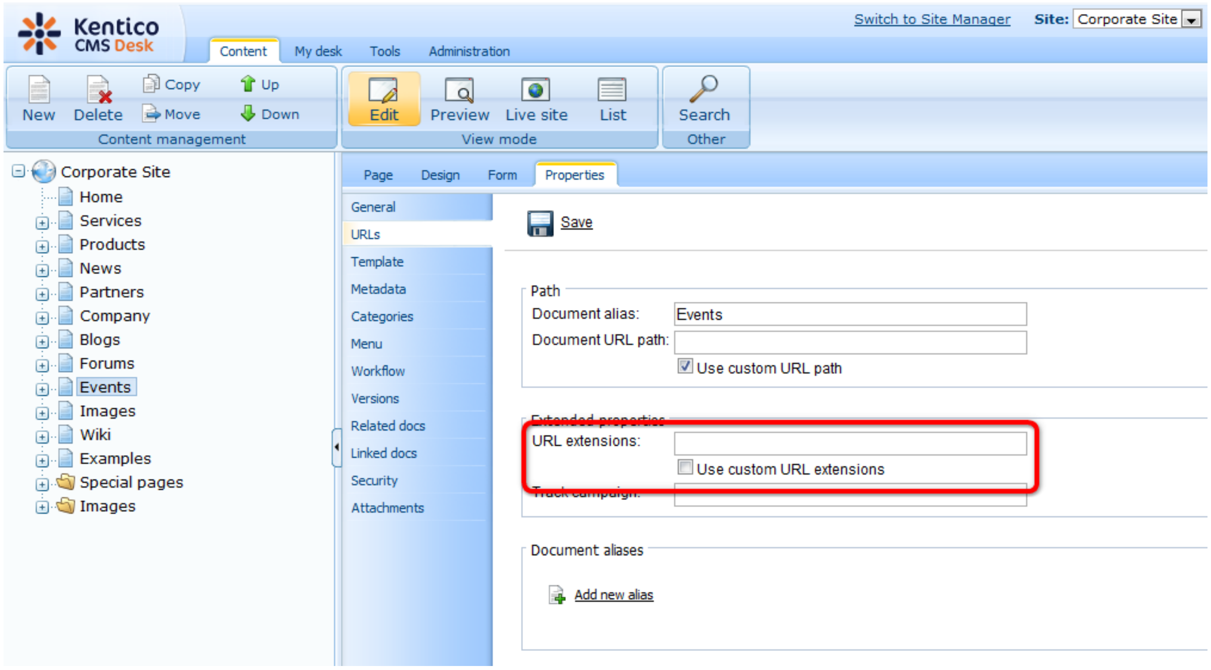
- The first extension is used as the default one and the links will be generated with it in the browser.
- Other extensions are entered divided by semicolons (;). Pages can be accessed through URLs ending with all entered extensions.
- If you use a semicolons without any extension in front of it, just as in the middle of the sample entry above, extension-less URLs can be achieved.

Document-level extensions settings

Apart from the global settings described above, document extensions under which the document can be accessed can also be set separately for each document. The default extension with which the pages are **rendered** in the browser is always taken from the **global** settings.

1. Select the document from the content tree.
2. Switch to its **Properties -> URLs** tab.
3. Enable the **Use custom URL extensions** check-box.
4. Enter the required extension(s) using the same rules as described above.

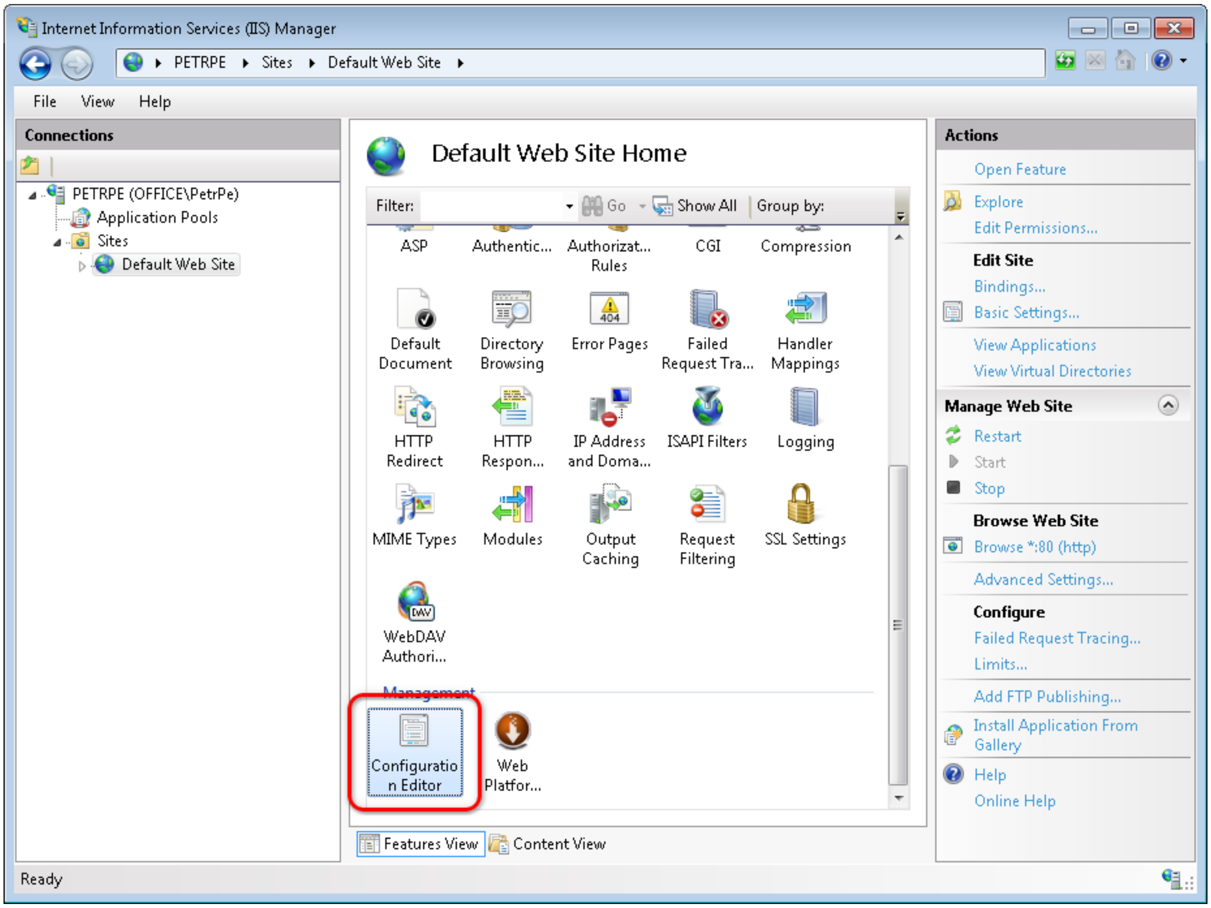
Please note: Even if the **Use custom URL extensions** option is disabled, files (**cms.file** documents) can be accessed under their physical extensions.



3.4.11.5 Lock violation on IIS7

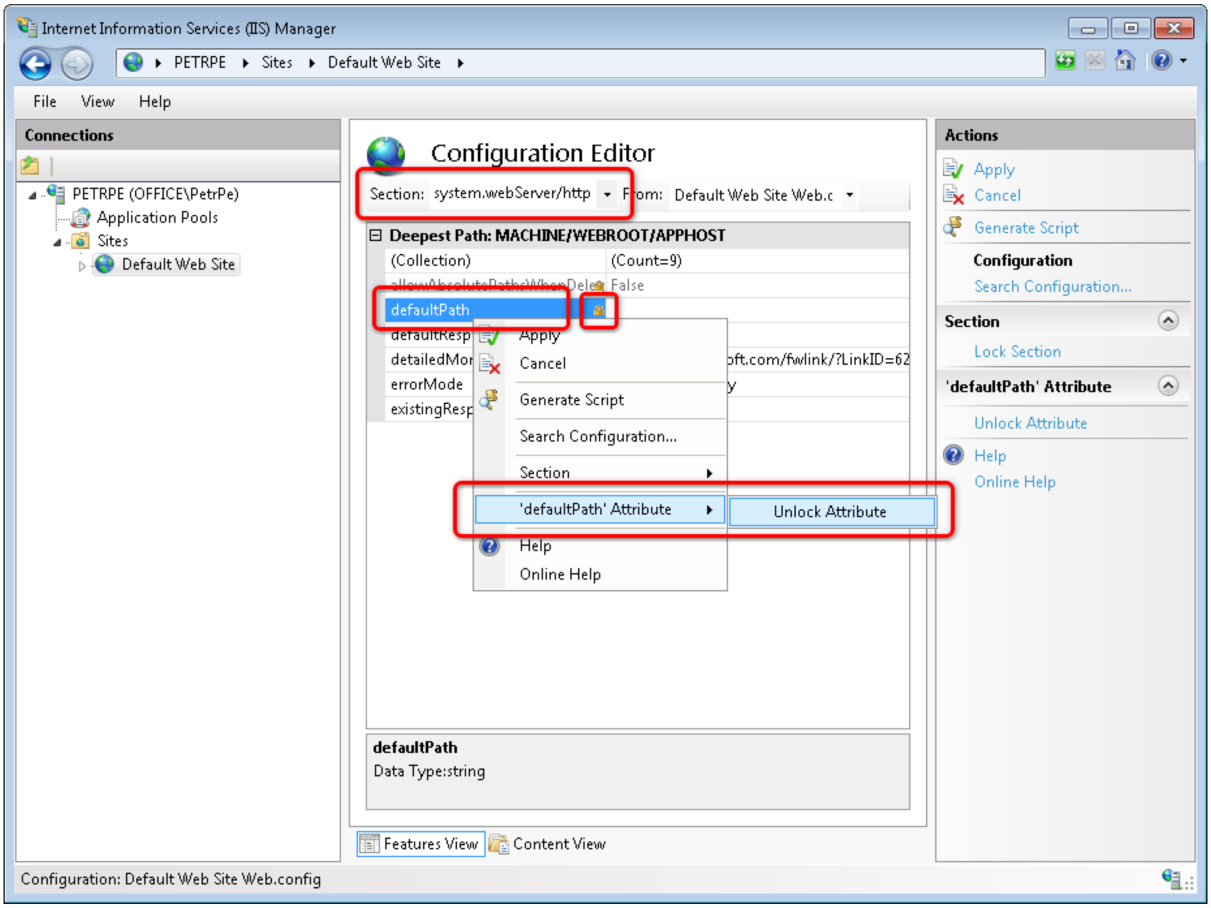
When configuring your IIS 7 (or later) to allow custom extensions or extension-less URLs, especially if you're running Kentico CMS in a virtual directory, you may receive the **Lock violation** error message. This typically doesn't allow you to specify the path settings in step 4 of the instructions for using cmspages/handler404.aspx described on this page.

The reason is most probably a locked **defaultPath** attribute in the **httpErrors** section. You can check and unlock it in **IIS Manager**. Select your site (IIS site) and open **Configuration Editor** (it is included in the *Management* section in the standard installation of IIS 7.5; if you are using IIS 7, you can download and install it as a part of the IIS7 Administration Pack).



In Configuration Editor, choose **webServer/httpErrors** from the **Section** drop-down list.

If there is a **lock icon** next to the **defaultPath** attribute name, right-click on the attribute name and select the **'defaultPath' Attribute -> Unlock Attribute** action from the context menu. If the option is missing in the context menu, you will probably have to unlock it on a higher level in the IIS tree, i.e. on the parent site or on the root of the server.

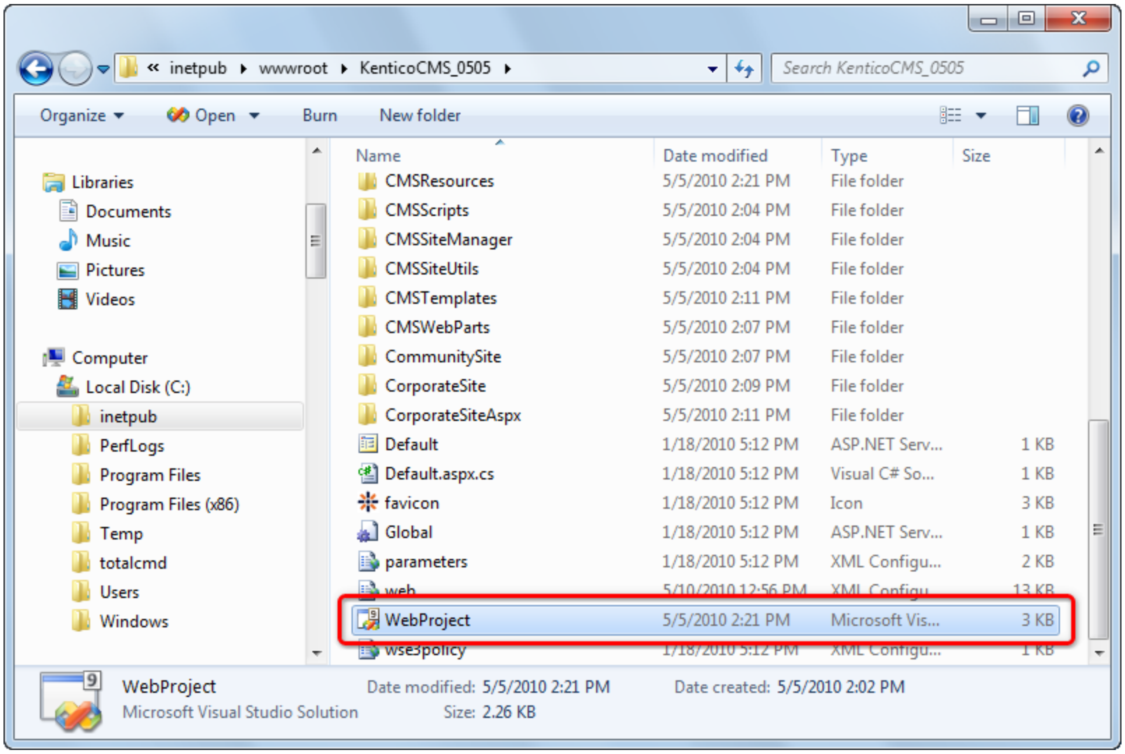


Click **Apply** to save the changes and from now, the *Lock violation* errors shouldn't appear.

3.5 Visual Studio integration

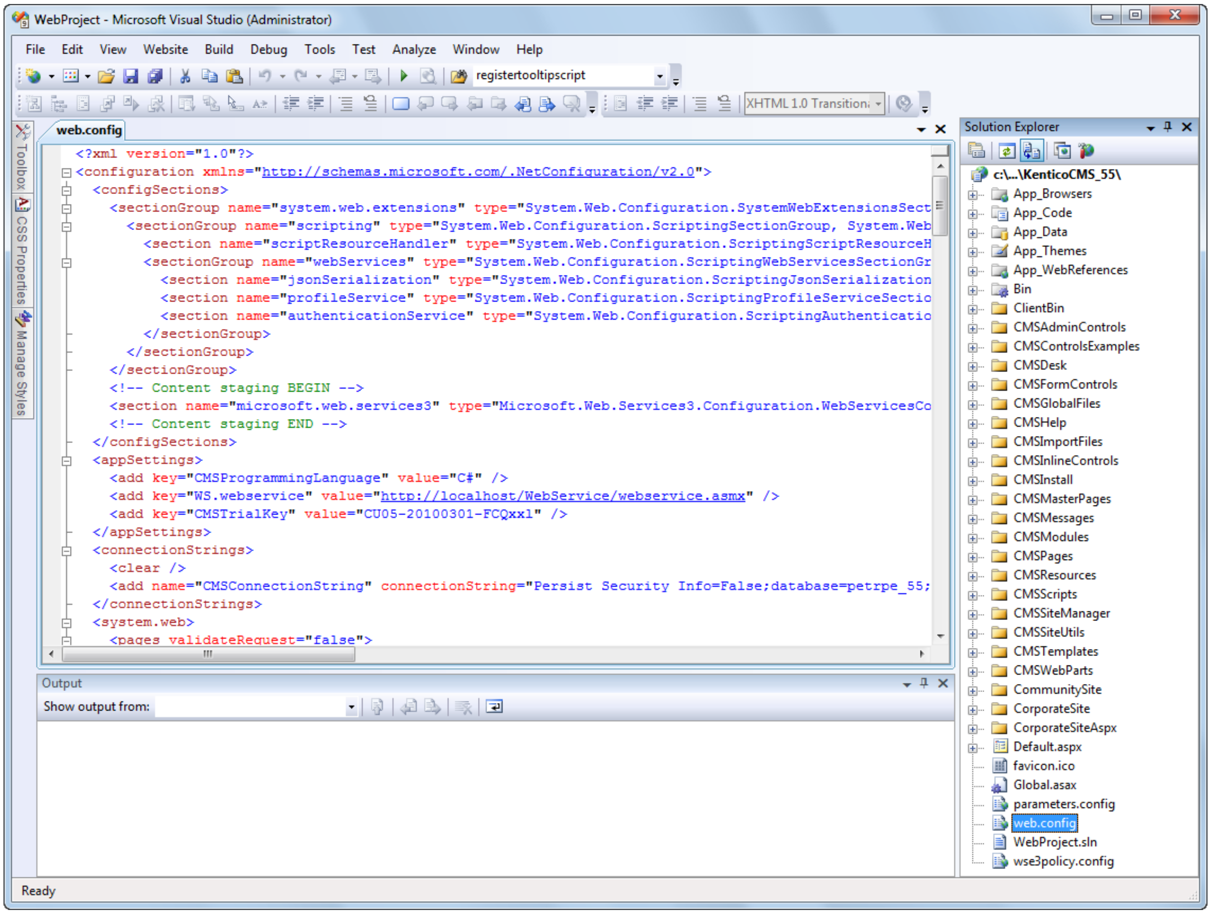
3.5.1 Opening the project

After you create a new Kentico CMS web project, you can open it in Visual Studio by double-clicking the WebProject.sln file in Windows Explorer:



If this option doesn't work, you can start **Visual Studio** and choose **File -> Open -> website** in the main menu and navigate to the folder which contains the *WebProject.sln* file.

The project looks like this:



Making modification to the standard code

Although Kentico CMS is delivered with source code of the administration interface (CMS Desk, CMS Site Manager), it's recommended that you do NOT modify the default files to avoid problems when upgrading to a higher version (your changes may be overwritten by new code).

If you need to modify some dialog, note down its name and merge your modifications with the new version during the upgrade to a higher version of Kentico CMS.

If you need to modify some web part, create its copy and then modify it.

Source Code Options

There are two levels of source code:

- the **source code of the website**, administration interface and web parts - this source code is delivered with every license (even in the trial version)
- the **full source code** of all libraries, including data layer, business layer and Kentico CMS Controls - this source code is only available as a part of the 1 website Ultimate or 1 Server Ultimate licence with Source Code.

You're allowed to modify the source code you receive (in both options) and deploy the modified version into production environment, provided you meet all other licensing conditions.

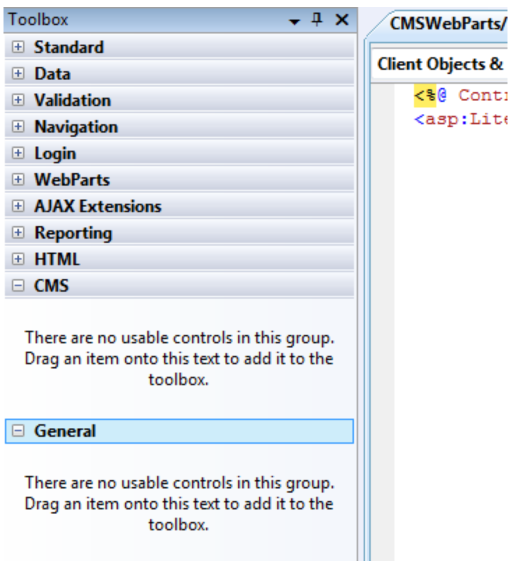
3.5.2 Adding Kentico CMS Controls to the Toolbox

Before you start using Kentico CMS Controls in your ASP.NET project, you need to add the controls to the **Toolbox**:

1. Open the website project in Visual Studio and open some ASPX page.

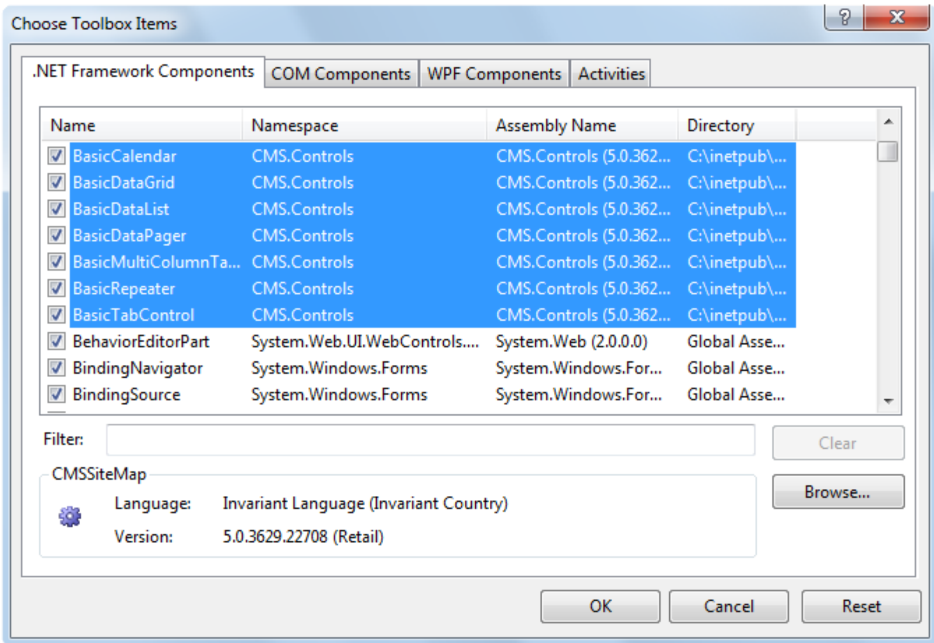
2. Right-click the **Toolbox** and choose **Add tab** from the context menu.

3. Type the name of the new tab (e.g. CMS) and press Enter:

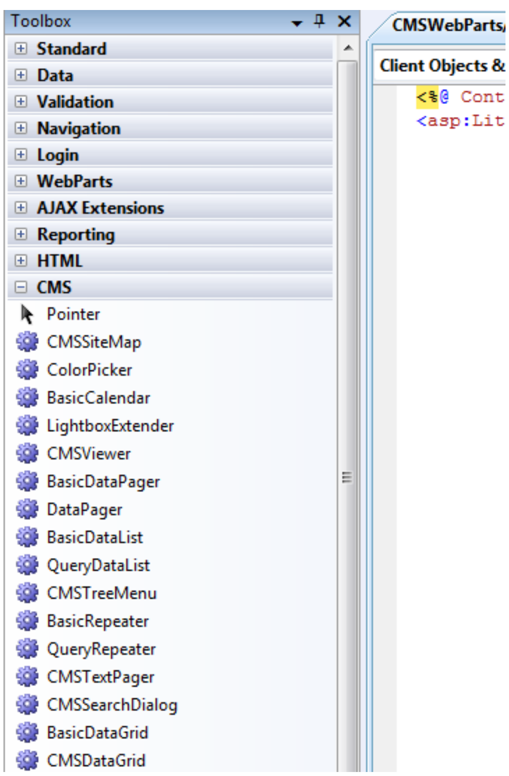


4. Right-click the new tab and choose **Choose items...** from the context menu.

5. In the **Choose Toolbox Items** dialog, click **Browse** and locate the **CMS.Controls.DLL** library in the **bin** folder under your website. Click **Open** and then click **OK**.

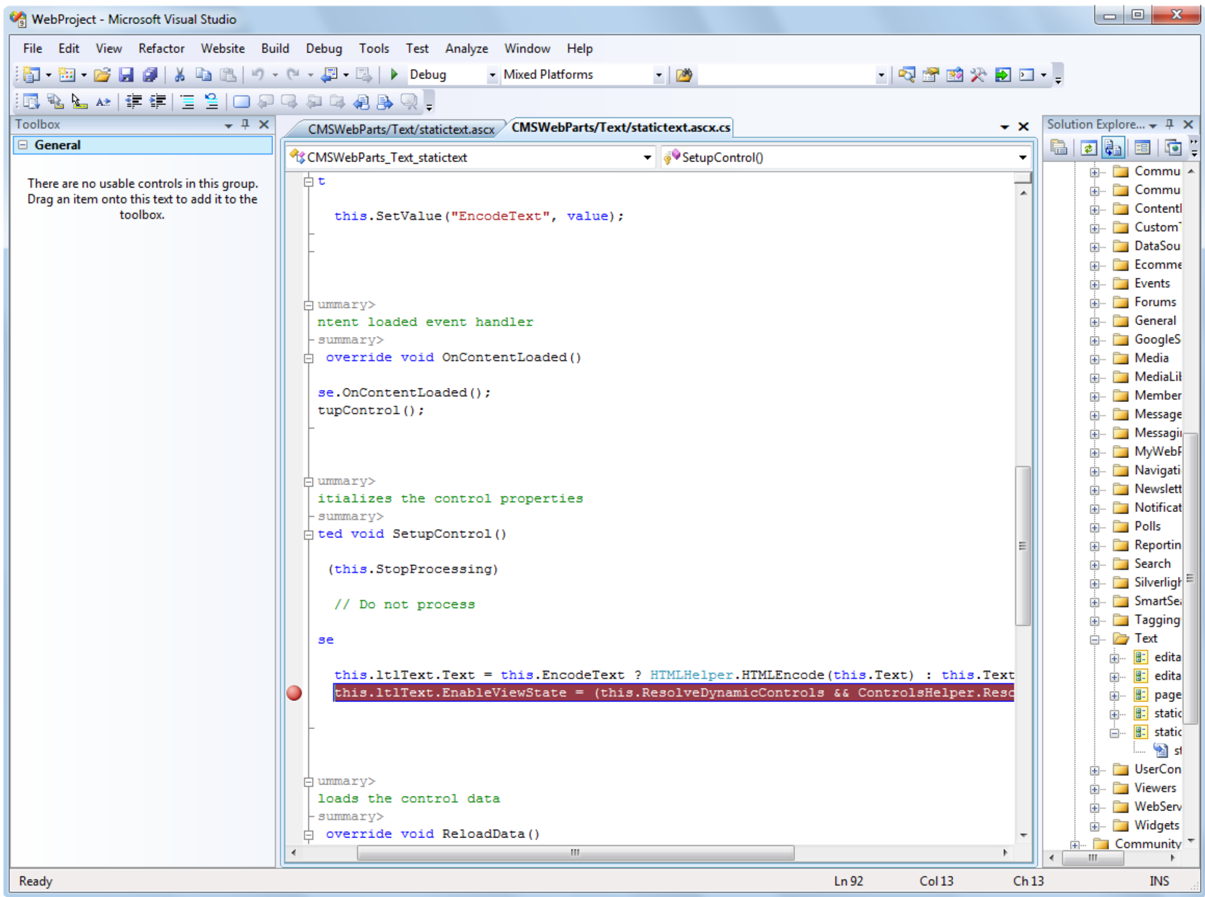


6. The controls are now added to the Toolbox, as you can see in the following screenshot. Now you can easily drag and drop the controls on your Web form.

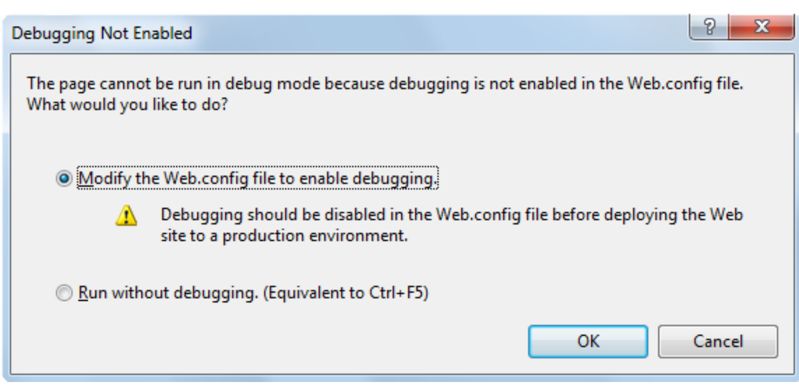


3.5.3 Debugging

If you're adding custom code using Visual Studio into Kentico CMS, you can easily debug it in Visual Studio as you're used to since Kentico CMS is a standard Visual Studio application. Simply click on the left next to your method or command and it will create a red breakpoint:



Now choose **Debug -> Start debugging** in the main menu or press **F5**. The website starts and you can track the code flow. You may get a message like this:



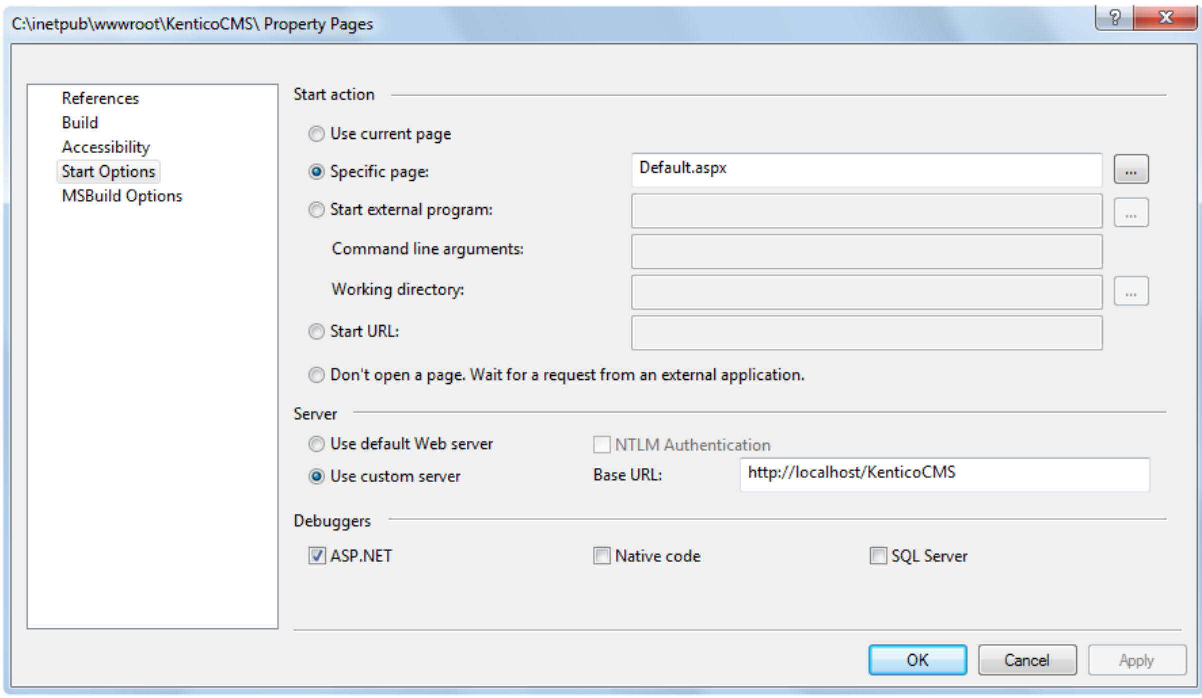
You will need to choose to **Modify the web.config file to enable debugging** and click **OK**. It's recommended that you disable debugging before deploying the website to a production environment by setting the value

```
<compilation debug="true" .... />
```

to **false** in your *web.config* file for better performance.

Debugging using IIS website

The debug mode starts in built-in web server by default. You can change this in the **website -> Start** options dialog by choosing the Use custom server option like this:



Please note that you need to be local administration and the website or virtual directory must be configured for both anonymous and Windows authentication (see Additional configuration tasks -> Creating a virtual directory for details).

Debugging from within Kentico CMS UI

Kentico CMS provides certain debugging possibilities directly from within its user interface in **Site Manager -> Administration -> System -> Debug**. Further information on this topic can be found in the Development -> Debugging chapter of this guide.

3.5.4 Pre-compilation (Publish function)

If you want to pre-compile the website before placing it on the live server, you can use Visual Studio's Publish function. It allows you to compile the source code into assemblies. This provides several advantages:

- faster application start (no compilation is required)

- intellectual property protection - if someone gets your website code, they will not be able to read the source code easily
- better security - the code cannot be easily modified by potential hacker

However, there's also a disadvantage: if you compile the website, you will not be able to use some of the Kentico CMS features:

- import of the website
- New site wizard.
- You cannot add code in the Code tab of web parts (the Code tab is an obsolete feature as of Kentico CMS 3.0, it is mentioned here due to backward compatibility reasons).

If you run a pre-compiled website, you will get the errors like:

The file '/CMSTransformations/cms/event/preview.ascx' does not exist.

This means that the system requires a virtual object, but the **VirtualPathProvider** is probably not running and cannot provide the system with that object.

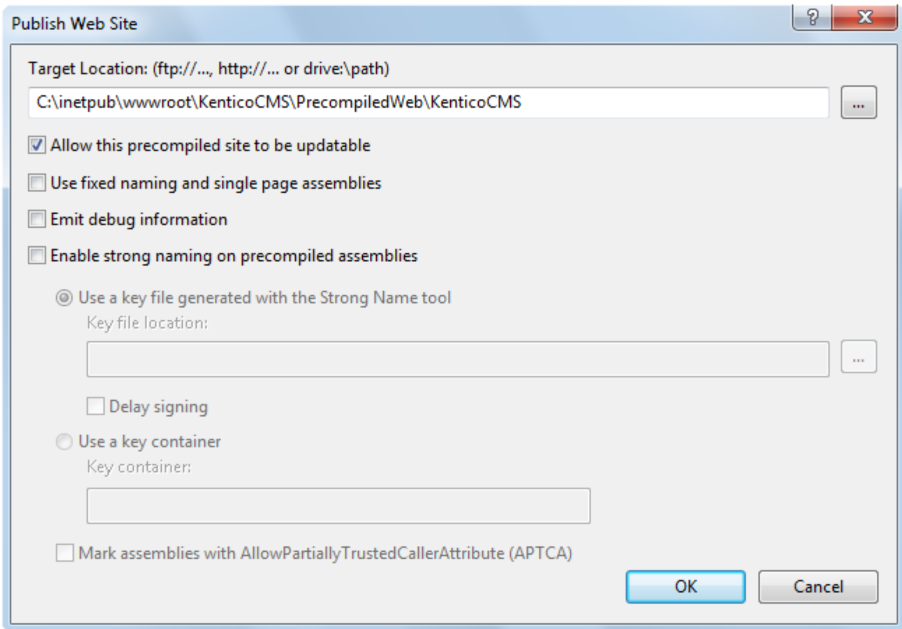
Pre-compiling the website

Please note that for building the pre-compiled website, you have to rename **Microsoft.Web.Services3.dll.rename** file to **Microsoft.Web.Services3.dll** in the ~*Bin* directory.

Since the virtual provider cannot run in a pre-compiled website, you need to store the physical files to the system before compilation. You can save all the virtual objects to the file system in **CMS Site Manager -> Administration -> System -> Deployment interface by clicking the button “Save all virtual objects to disk”**.

Please note that these files are just copies of the actual virtual object and will be used by the system only if the VirtualPathProvider cannot start. Also, the changes to the objects through the administration interface will not affect these files until you save all the objects to the disk again.

Then, you can use **Build -> Publish** item in **Visual Studio** main menu and pre-compile the website. You will see a dialog like this:



You need to enter the path outside your current web project where the compiled version will be placed. You can choose between two compilation modes using the **Allow this precompiled site to be updatable** checkbox:

- **Allow the site to be updatable (checked) -** the code behind and app_code classes are compiled, but ASPX files are stored in their original source code form.
- **Do not allow the site to be updatable (unchecked) -** the code behind and app_code, as well as ASPX files are compiled and the website contains only empty "stub" ASPX files without any code.

We generally recommend that you do not allow the site to be updatable, since it provides the best performance.

Limitations

In the precompiled website, the VirtualPathProvider is stopped automatically. When VirtualPathProvider is stopped, **you cannot edit transformations and layouts through the user interface** without saving them on the disk again.

You may need to copy the database to the server using the standard **backup/ restore** operation since the compiled website cannot be used for SQL Server database installation. Other options are:

- to install a non-compiled website on the server first, go through the setup wizard and then replace the non-compiled files with compiled ones, while keeping the web.config file as is.
- to install Kentico CMS website locally and run the database setup against the remote SQL Server on the live server.

In the portal development model, you cannot use **custom web part code** (**Web part properties –> Code tab**). If you need to add custom code on the Code tab and run the website in the compiled version, you need to create user controls, place web parts to the user controls and add your custom code to the web parts. Then, you can place the user controls to the page using the General/User control web part.

Testing the website before compilation

If you wish to check if your website runs without using **VirtualPathProvider** (to simulate the precompiled environment), you can disable the provider by adding the following key into the appSettings section of your web.config file:

```
<add key="CMSUseVirtualPathProvider" value="false" />
```

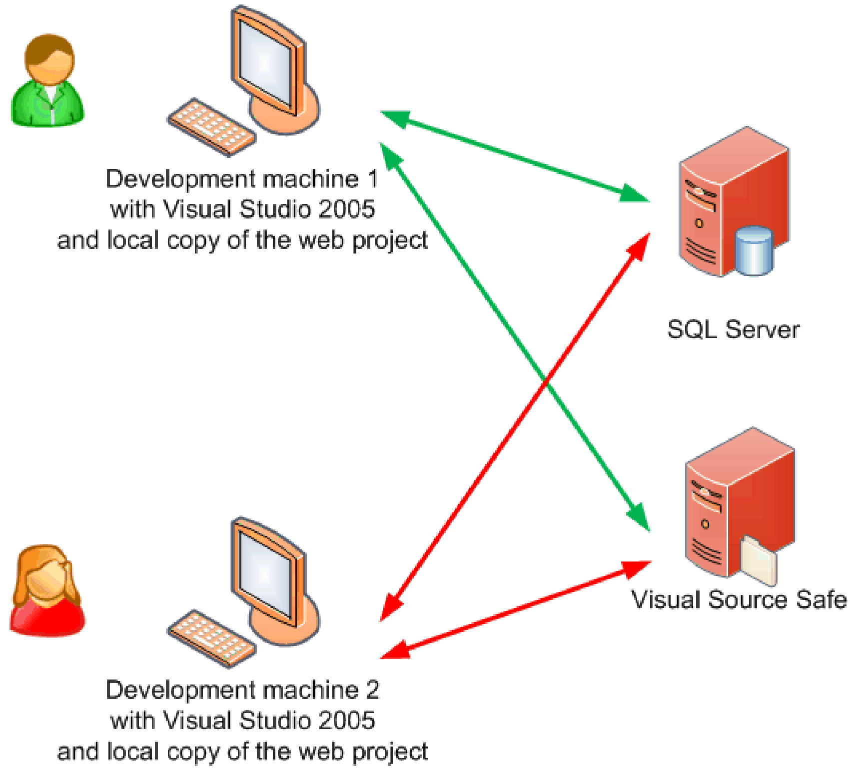
If your system runs with VirtualPathProvider disabled, you should be able to run it after the precompilation or deployment process.

Related topics: Additional configuration tasks -> Configuration for Medium Trust environment, Installation procedure -> Deployment to the live server, Additional configuration tasks -> Installation on shared hosting server

3.5.5 Visual Source Safe and Team Development

Kentico CMS can be used in team development environment as any other website project in Visual Studio. You can also use Microsoft Visual Source Safe (VSS) as you're used to: you simply add the website to VSS and then you need to check out/check in the files you want to modify.

In this case, all developers have their local copy of Kentico CMS installed, but they use the same VSS code and the same database as shown on the following picture:



Synchronization of memory objects between development machines

Kentico CMS caches some system objects (such as transformations, templates, etc.) in memory. It means that the memory on multiple development machines may not be synchronized and the developers may not see the latest version and they may even overwrite the work of other developers. That's why we recommend you to synchronize the memory objects between development machines using the Web farm synchronization module.

Team Development without Visual Studio

If the developers do not modify the source code and use the portal engine development model, they do not even need to have the local copies of the web project and they do not need to use VSS. In this case, they can install a single instance of Kentico CMS on their web server and develop the website through the browser-based interface.

3.5.6 Opening a VS2005 project in VS2008

If you were using Visual Studio 2005 for your web project and wish to convert to Visual Studio 2008, all you need to do is:

1. Start **Visual Studio 2008**.
2. Click **File -> Open website...** .
3. Choose the folder with your web project on the disk and click **OK**.
4. If you're asked if you wish to upgrade the project to a newer version of .NET Framework and Visual

Studio, click **Next** and go through the wizard.

Compilation error

You may receive a compilation error saying there are different versions of the **System.Web.Extensions.dll** library in the Global Assembly Cache (GAC) and a temporary folder. In this case, you need to locate the file **bin/System.Web.Extensions.dll** in your web project and delete it.

3.6 Troubleshooting installation issues

3.6.1 Overview

You may encounter various issues during the installation. The following chapters may help you sort them out.

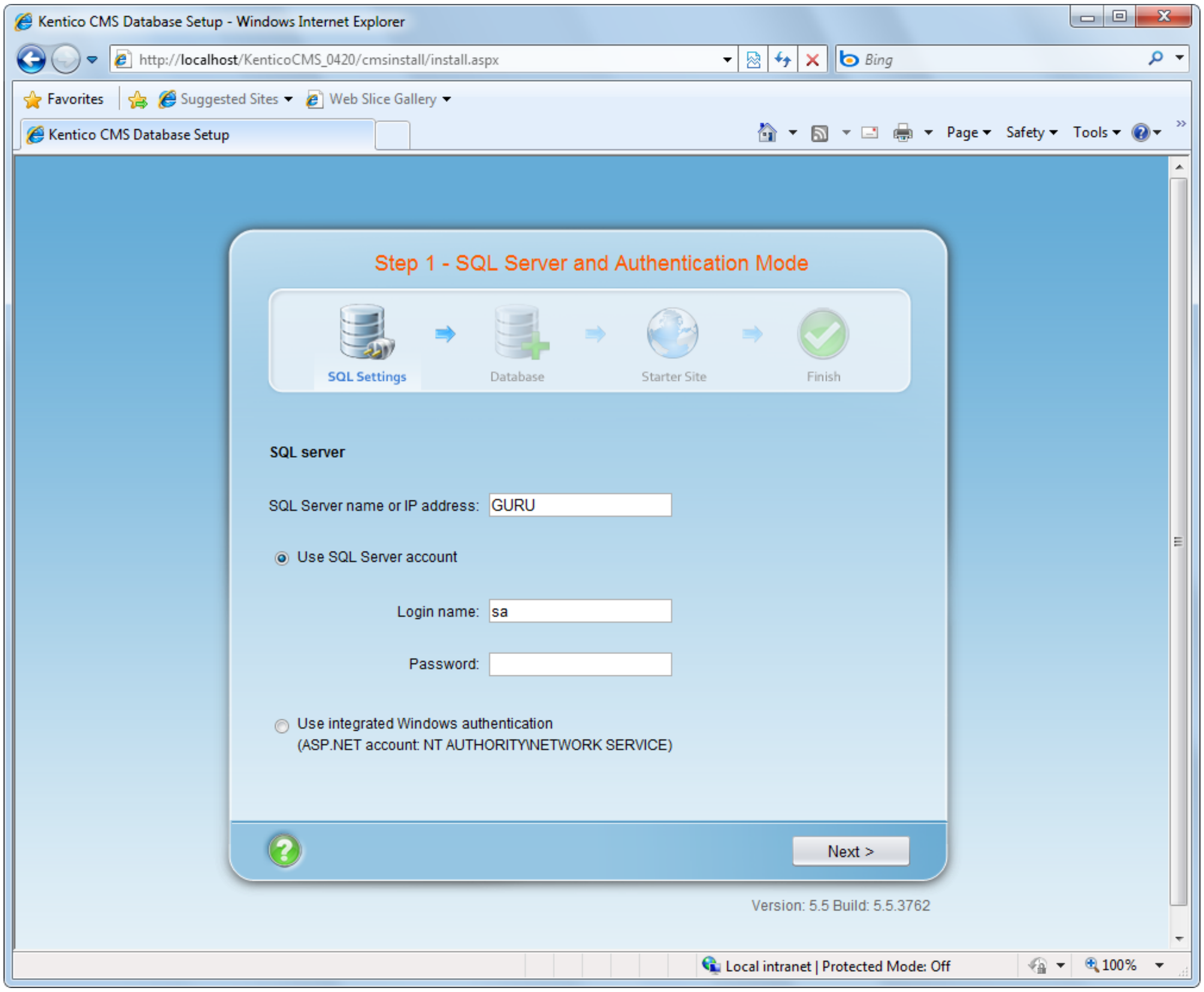
You may encounter problems in following areas:

- SQL Server connection problems
- ASP.NET not working on Windows Server 2003
- Internal server error when using Integrated pipeline mode
- Disk permissions problems

Click one of the links to learn how to solve the issues.

3.6.2 SQL Server connection problems

You may encounter problems when entering the database connection details in the first step of database setup:



Error 1: Establishing connection to the server

Error message:

An error has occurred while establishing a connection to the server. When connecting to SQL Server 2005, this failure may be caused by the fact that under the default settings SQL Server does not allow remote connections. (provider: Named Pipes Provider, error: 40 - Could not open a connection to SQL Server)

Troubleshooting:

1. Make sure the SQL Server name or IP address is correct. In some cases, using one of the following values may help:

- your computer name

- localhost
- 127.0.0.1
- (local)

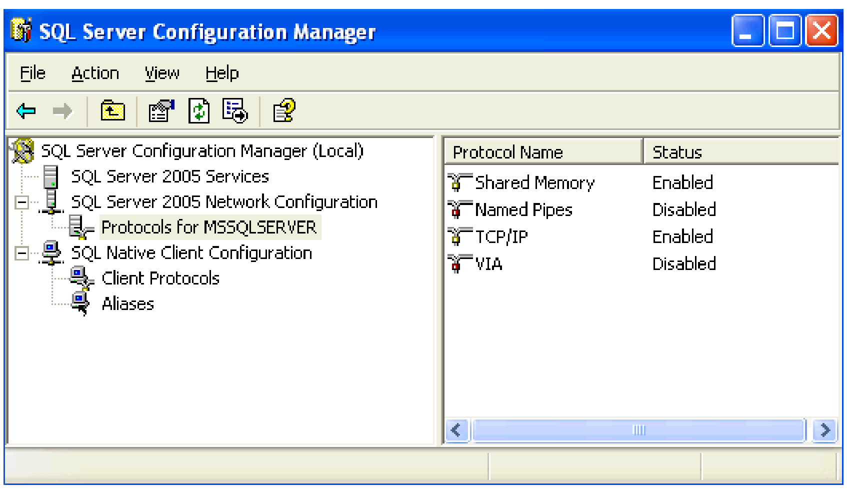
2. Make sure the server has Microsoft SQL Server 2005 or 2008 installed and running.

3. Make sure you are using the appropriate instance of the SQL Server in case you are using different instances of SQL Server. The instance name must be entered as myserver\myinstance (please note there's a backslash ****).

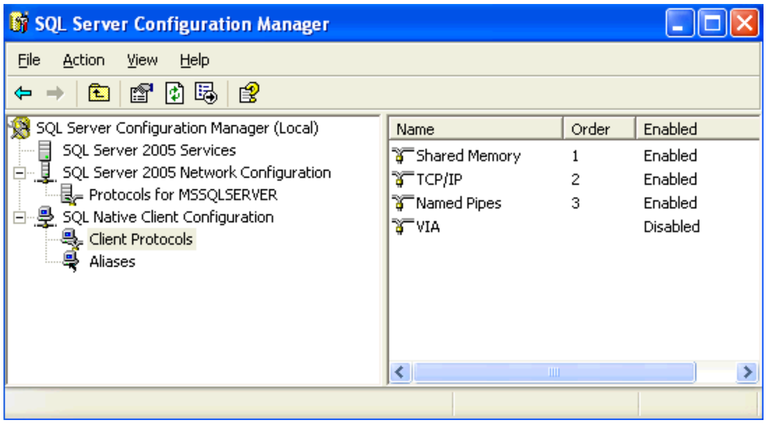
4. If you're using Microsoft SQL Server Express 2005 with default installation settings, the correct server name is **.\sqlexpress** or **computername\sqlexpress** .

5. Make sure the access to the database server is not blocked by some firewall (the default port number for TCP/IP protocol is 1433).

6. If you're using **SQL Server 2005** (especially the Express Edition), some protocols are disabled by default. You may need to go to **Start menu -> All Programs -> Microsoft SQL Server 2005 -> Configuration Tools** on the computer where the SQL Server is installed and start **SQL Server Configuration Manager**. Then, go to SQL Server 2005 Network Configuration and enable the TCP/IP protocol:



7. You may also need to enable the TCP/IP protocol in the **SQL Native Client Configuration -> Client Protocols** section:



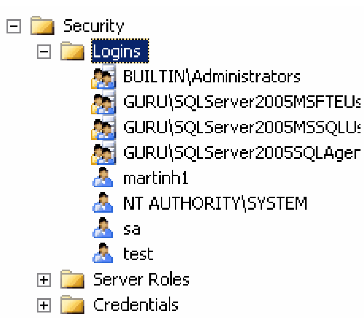
Error 2: Login failed for user 'xy'

Error message:

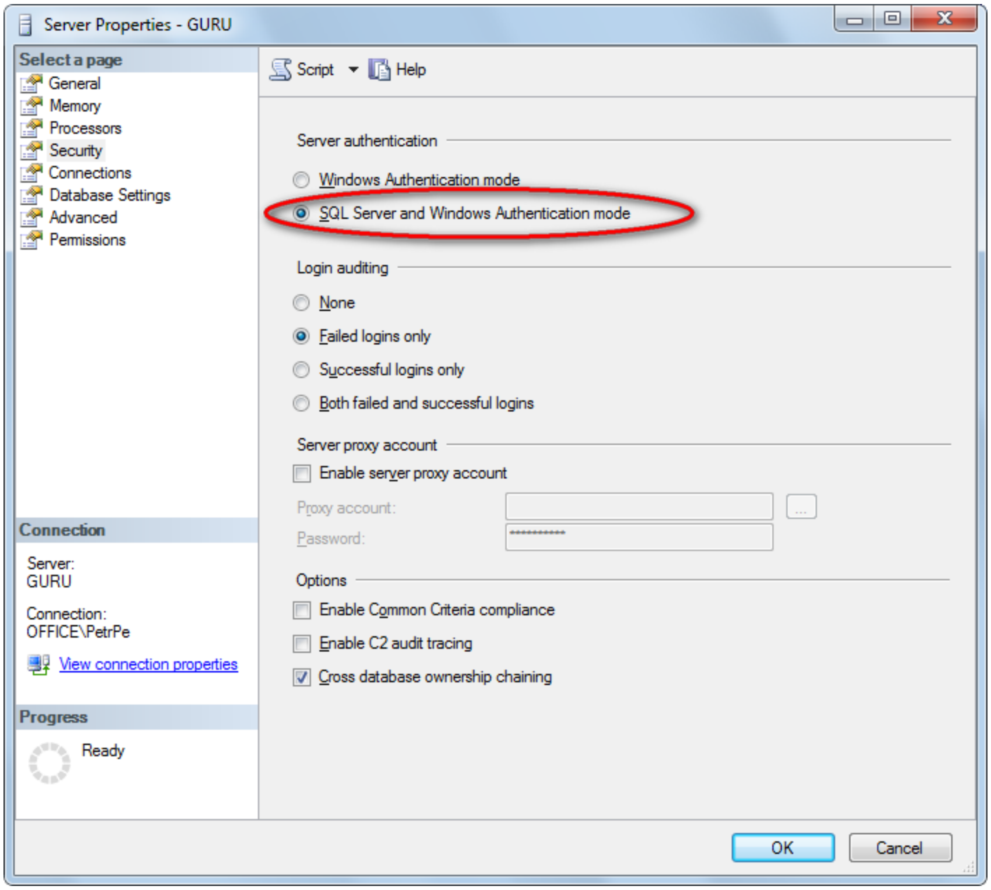
```
Login failed for user 'xy'
```

Troubleshooting for SQL Server account

If you're using SQL Server account with password, make sure you are using a valid user name and password. The login must be created on the server, it must be enabled and permissions to connect to the server must be granted to it. You can check the user account in Enterprise Manager/SQL Server Management Studio -> Server -> Security -> Logins:



Also, check the **Server Properties -> Security** dialog in Enterprise Manager/SQL Server Management Studio and make sure your server supports **SQL Server and Windows Authentication mode**:



Troubleshooting for Windows Authentication account

If you're using Windows Authentication account, the situation may be a little more complex and may require you to contact your network administrator. ASP.NET applications run under some particular local or domain account. This current account is displayed on the screen:

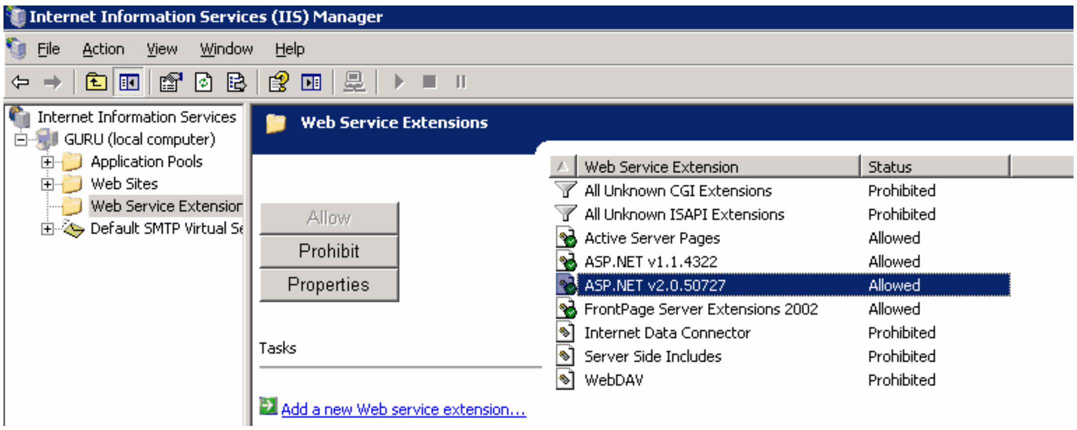
This account must have its own login with Windows authentication in the SQL Server. You can create the login in **Enterprise Manager\SQL Server Management Studio -> Security -> Logins** and grant appropriate permissions on the server to it. If you SQL Server is located on a different machine than your web server, you may need to configure your web application so that it runs under some domain account, rather than local account so that you can the login in the remote SQL Server.

If you do not succeed to configure Windows authentication, you may want to enable Windows and SQL Server Authentication on your SQL Server and use SQL Server account instead. You can learn more about SQL Server authentication in the **Troubleshooting for SQL Server account** section earlier in this chapter.

3.6.3 ASP.NET not working on Windows Server 2003

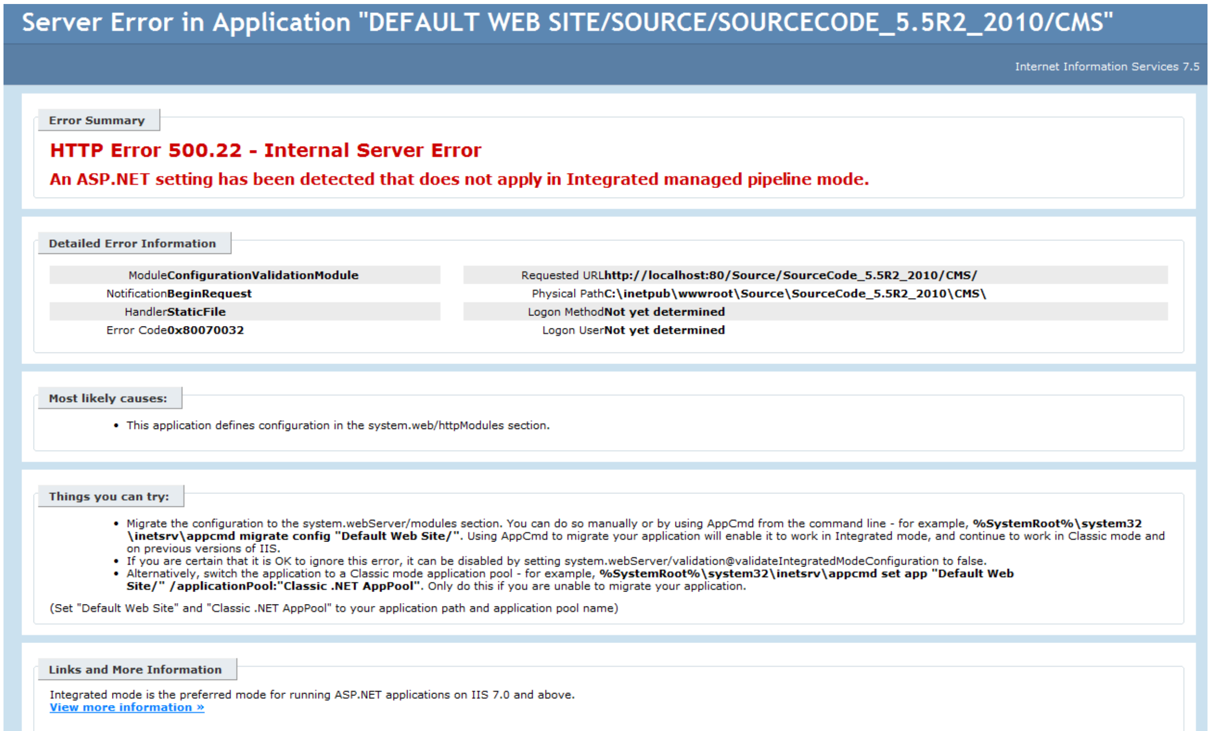
If you get the *404: Page not found* error or a similar error every time you request some ASPX page on your server and this is the first ASP.NET application installed and running on your server, it may be caused by configuration of Web Services Extensions on Windows Server 2003.

Go to **Control Panel -> Administrative tools -> Internet Information Services (IIS) Manager**, click **Web Services Extensions** and make sure that **ASP.NET 2.0** (or higher depending on the version you use) is **Allowed** (IIS 6 may display the **ASP.NET version as 2.0.50727** even if you have a higher one installed and registered) as shown in the following screenshot:



3.6.4 Internal server error when using Integrated pipeline mode

When installing Kentico CMS on a Remote server using a project created by the web installer or when running from a source code package, you may encounter an Internal Server Error while starting the Database setup. This occurs if the application pool of your IIS application is set for .NET 4.0 and *Integrated* managed pipeline mode (on IIS 7.0 or higher).



It is caused by the fact that your **web.config** file is not configured for *Integrated* pipeline mode by default. The application pool mode can be selected in the web installer during the course of an installation to a local IIS server, which causes the correct web.config to be created, but this is not the case when creating a project for a remote server or when using a project from a source code package.

The solution is to replace the default web.config file of your project with one containing the necessary configuration. The appropriate files can be downloaded here:

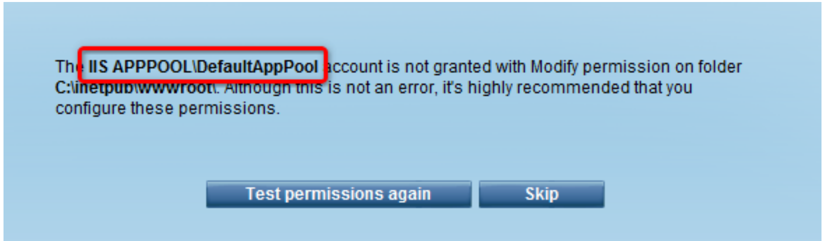
- Web.config for .Net 4.0 and Integrated managed pipeline mode
- Web.config for .Net 4.0 and Classic managed pipeline mode

3.6.5 Disk permissions problems

3.6.5.1 Disk permissions problems

Kentico CMS is able to perform most operations without writing to disk. However, there are situations when the web application needs to write to the disk for optimal operations or performance, such as importing/exporting a site or storing uploaded files in the files system (which is optional).

If you receive the error message depicted below, saying that the web application cannot write to disk, you need to grant the **Modify** permissions on the whole website folder to the appropriate user account.



User account of the web application

The web application runs under a user account that depends on your environment. Please note that the accounts listed below are just the default ones, they may be different in your environment. However, the **name of the account is always displayed with the error message**, as highlighted in the screenshot above.

1. On Windows 7, the user account is the local **IIS_IUSRS** account by default.

2. On Windows Vista or Server 2008, the user account is the local **NETWORK SERVICE** account by default.

3. On Windows XP, the user account is the local **ASPNET** account (aspnet_wp) by default.

4. On Windows 2003, the user account is the local account **NT Authority\Network Service** by default.

5. If you're using **Visual Studio's** built-in web server, it is running under your account.

You can see the name of the user account under which the application runs in **Site Manager -> Administration -> System** dialog.

Choosing the component for directory operations

If you're running Kentico CMS under restricted trust level, you may need to use the managed component for directory operations (create/delete/rename directory). You can configure it by setting the following web.config parameter:

```
<add key="CMSDirectoryProviderAssembly" value="CMS.DirectoryProviderDotNet" />
```

If you're running Kentico CMS on a shared hosting server, some providers require that you use the non-managed methods for directory operations:

```
<add key="CMSDirectoryProviderAssembly" value="CMS.DirectoryProviderWin32" />
```

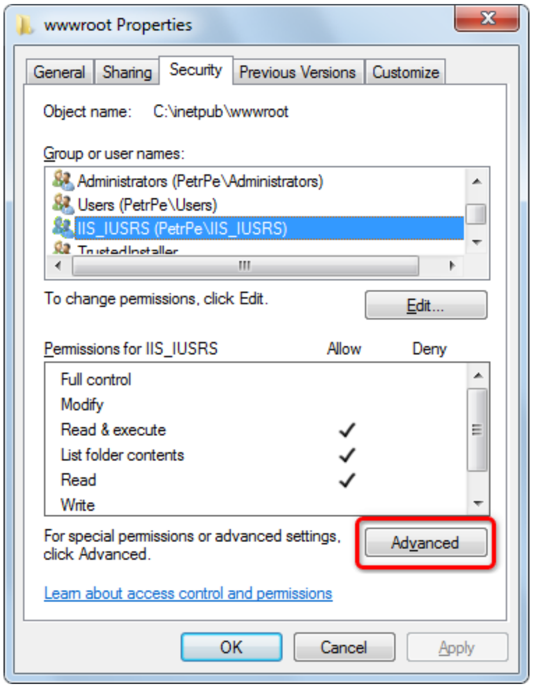
3.6.5.2 Solution on Windows 7

1. Open Windows Explorer, locate the folder with your website, right-click the folder and display its **Properties**. Choose the **Security** tab.

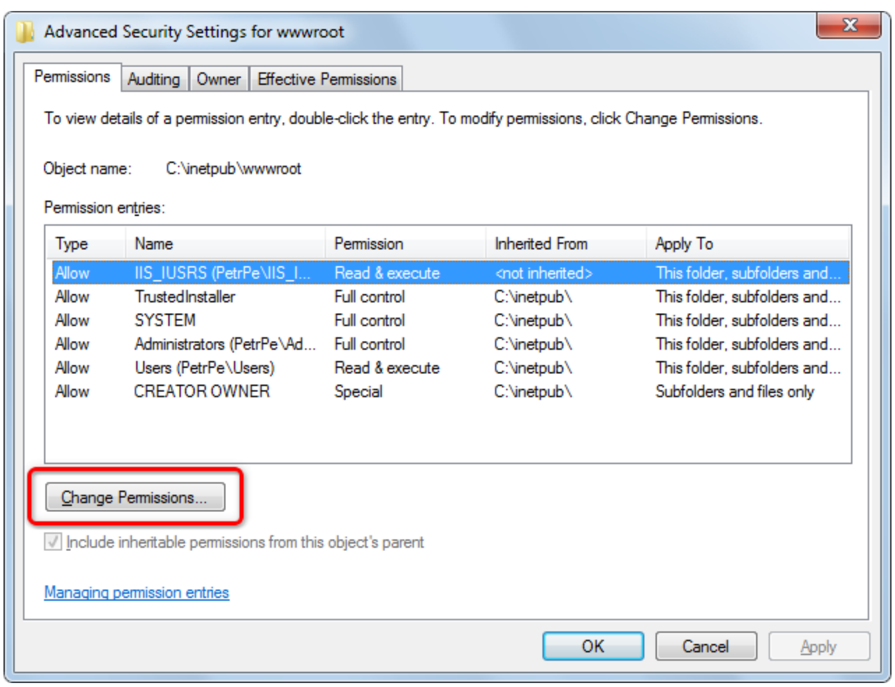
2. Verify that the account that you need to grant the permissions to (the name was displayed with the error message) is present in the **Group or user names** list. If not, click **Edit**, click **Add** in the pop-up

dialog and add the required account to the list. Close the pop-up dialog when finished.

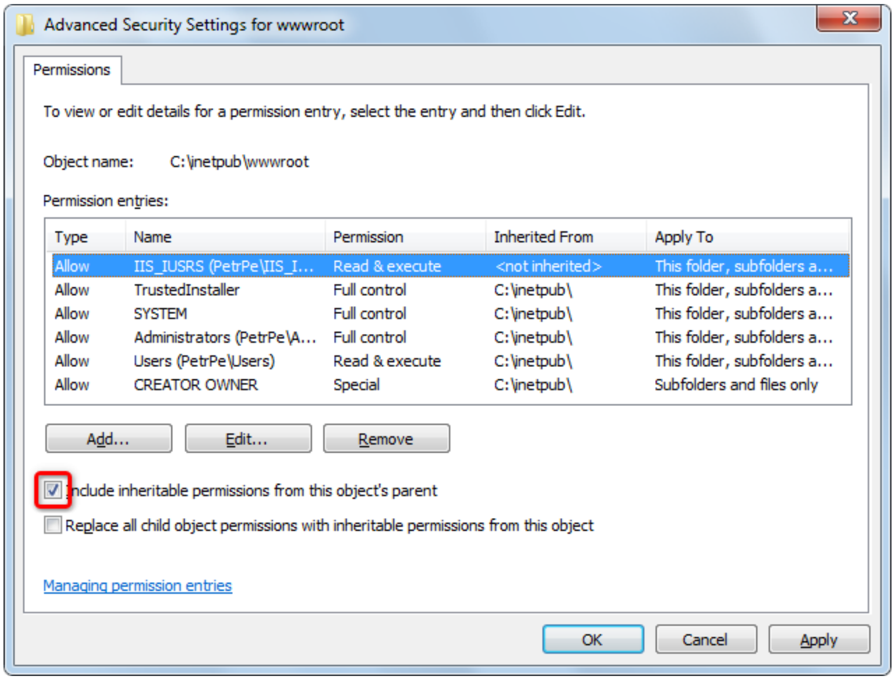
3. Back on the **Security** tab of the folder properties, select the appropriate account and click **Advanced**.



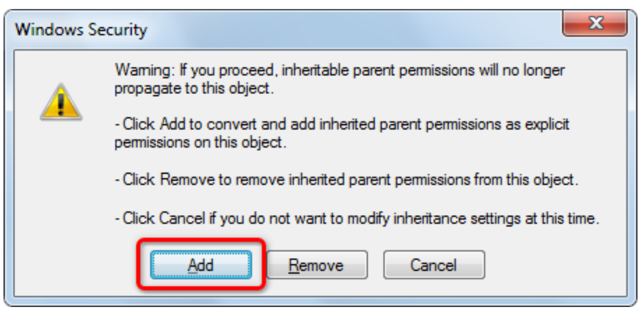
4. In the pop-up dialog, select the account again and click **Change Permissions**.



5. Disable the **Include inheritable permissions from this object's parent** option.

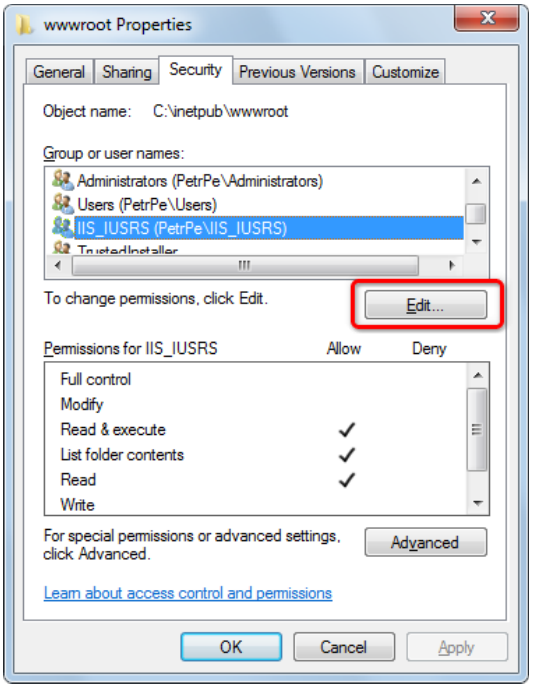


6. In the pop-up dialog, click **Add**.

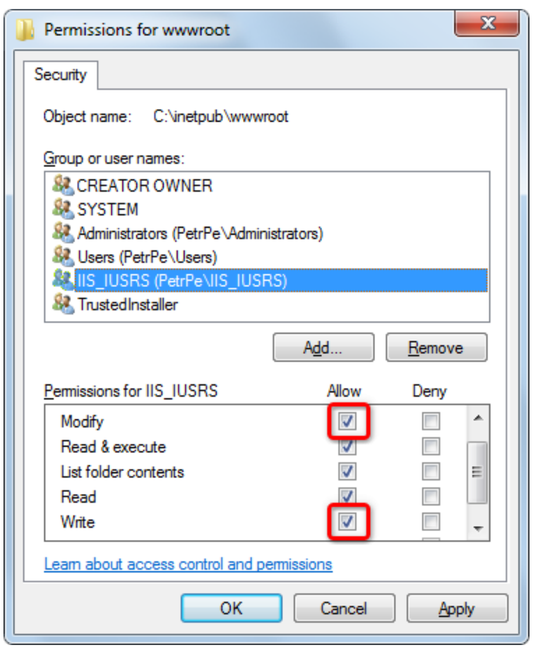


7. When the dialog closes, click **OK** to close the dialog under it. Click **OK** again to close the dialog under the previous one.

You are back in the folder properties dialog now. Select the account and click **Edit**.



8. Check the **Allow** check-box for the **Write** and **Modify** permissions and click **OK**.



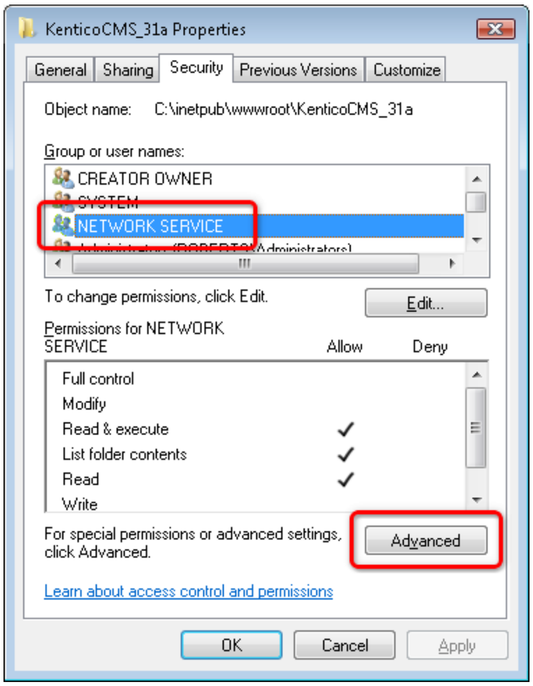
9. You have assigned the account with the required permissions. Kentico CMS should now be able to perform all disk write operations and therefore work correctly.

3.6.5.3 Solution on Windows Vista or Server 2008

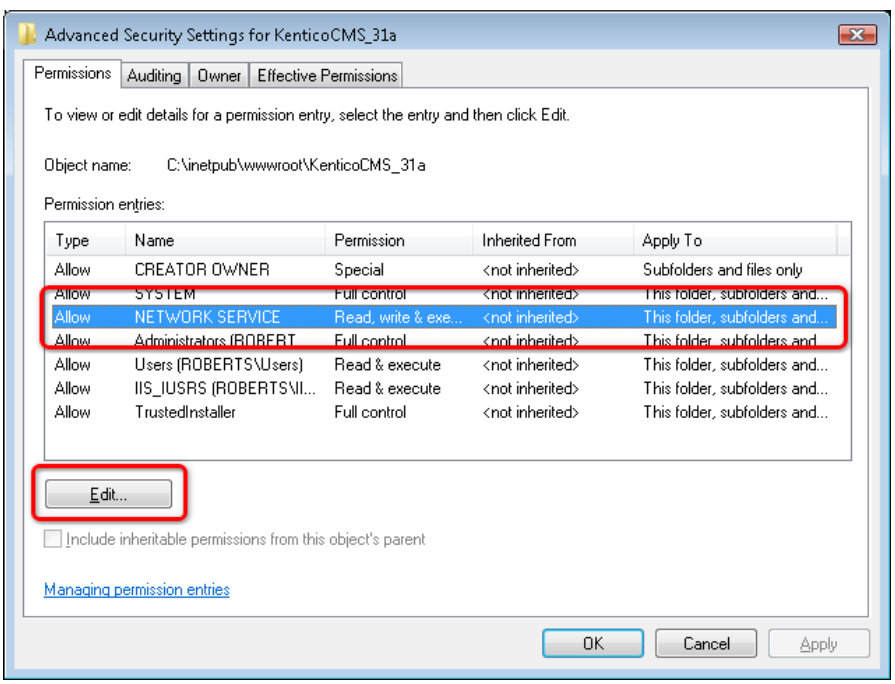
1. Open Windows Explorer, locate the folder with your website, right-click the folder and display its **Properties**. Choose the **Security** tab.

2. Verify that the account that you need to grant the permissions to (the name was displayed with the error message) is present in the **Group or user names** list. If not, click **Edit**. In the pop-up dialog, click **Add** and add the required account to the list. Close the pop-up dialog when finished.

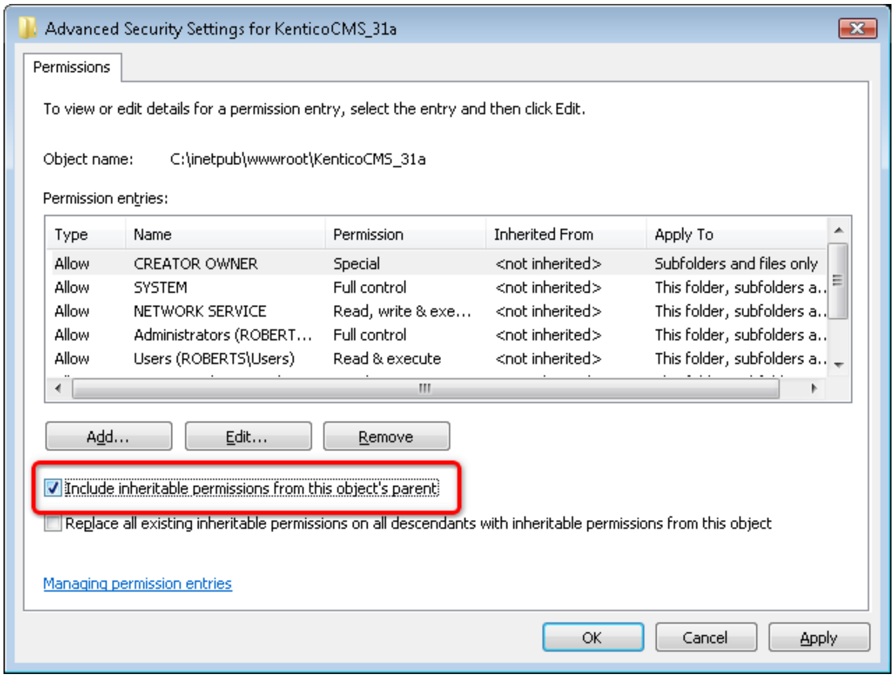
3. Select the required account and click **Advanced**.



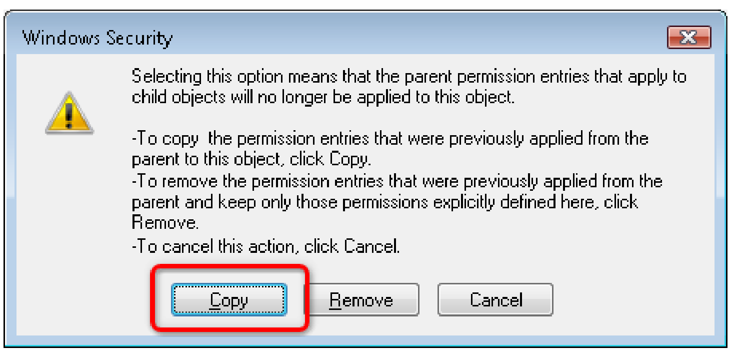
4. In the pop-up dialog, select the account again and click **Edit**.



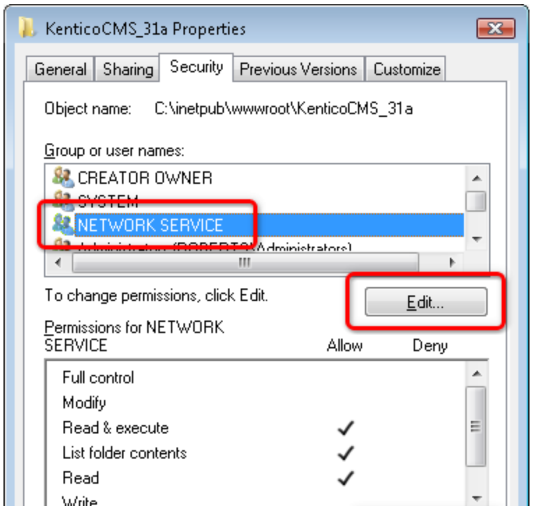
5. Disable the **Include inheritable permissions from this object's parent** option.



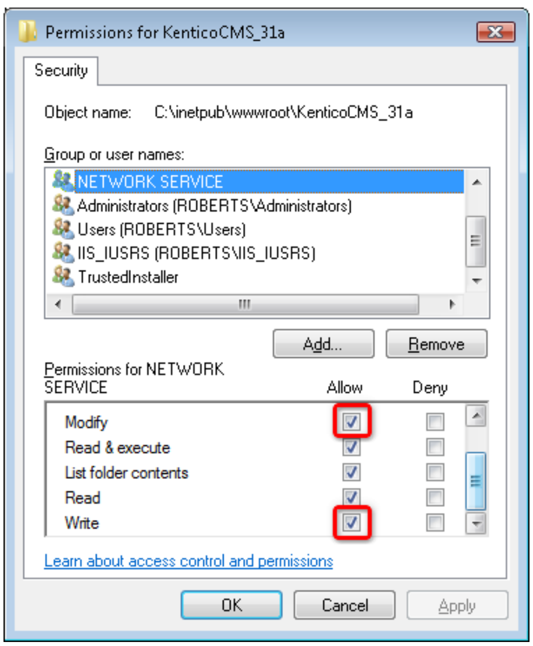
6. On the pop-up dialog, click **Copy**.



7. When the dialog closes, click **OK** to close the dialog under it. Click **OK** to close the dialog under the previous one. You are back in the folder properties dialog now. Select the account again and click **Edit**.



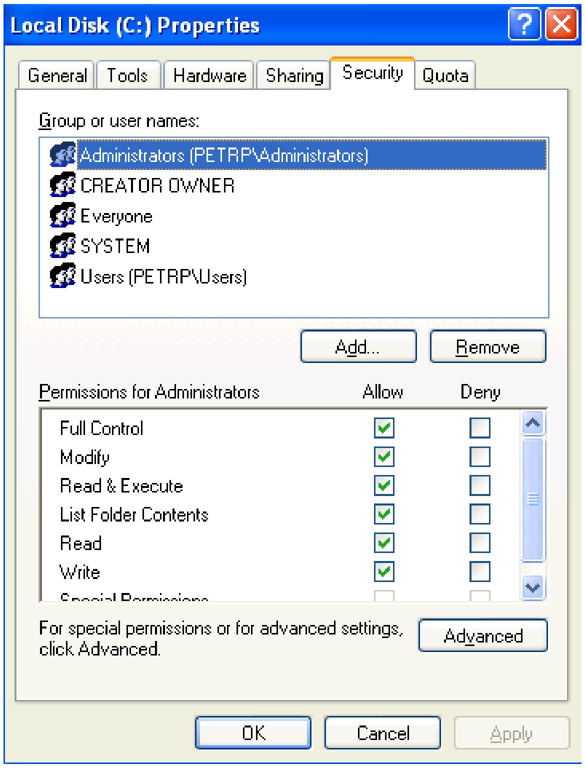
8. Check the **Allow** check-box for the **Write** and **Modify** permission and click **OK**.



9. You have assigned the account with the required permissions. Kentico CMS should now be able to perform all disk write operations and therefore work correctly.

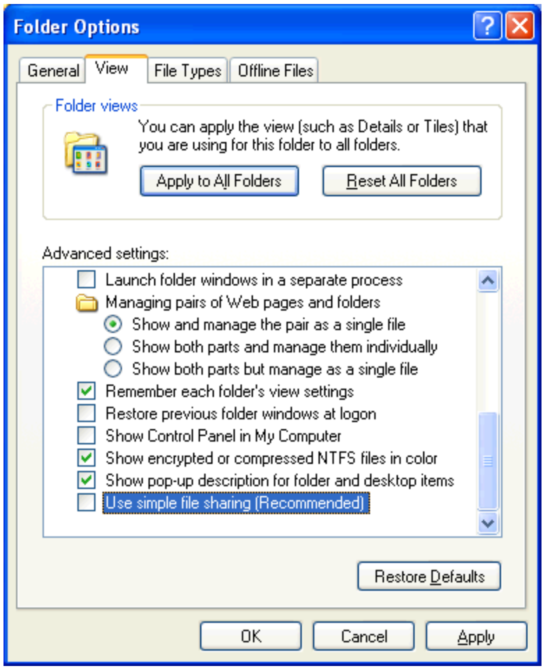
3.6.5.4 Solution on Windows XP

1. Open Windows Explorer, locate the folder with your website, right-click the folder and display its **Properties**. Choose the **Security** tab and click **Add...**.

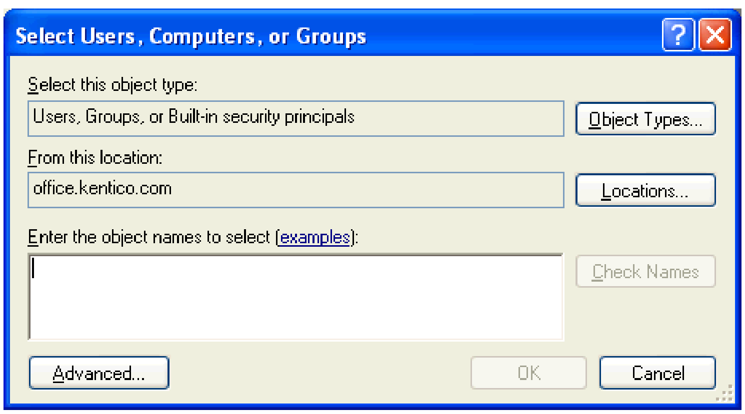


Missing Security tab in folder properties dialog

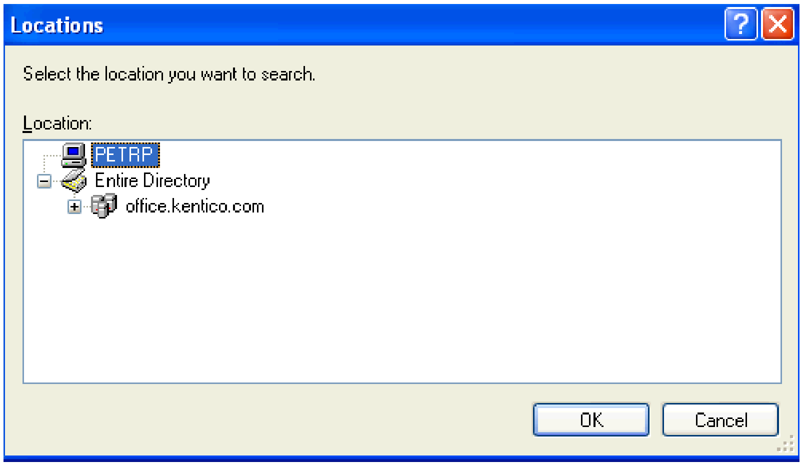
If you cannot see the Security tab, click **Tools -> Folder options** in the Windows Explorer main menu, choose the **View** tab and uncheck the **Use simple file sharing** box. Click OK. Now you should find the Security tab in the folder properties dialog.



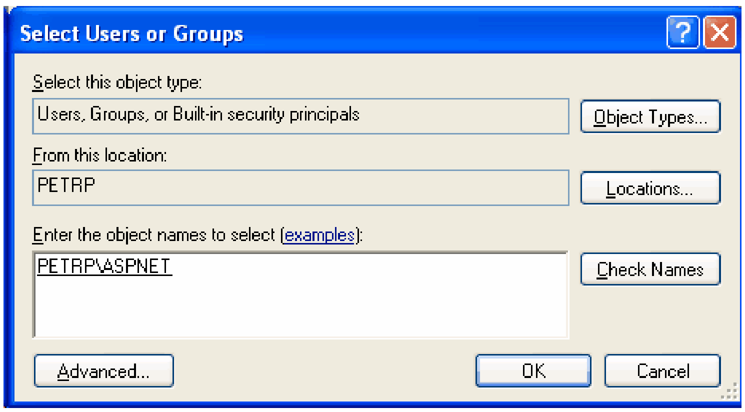
2. The **Select Users, Computers and Groups** dialog appears. Click **Locations...**.



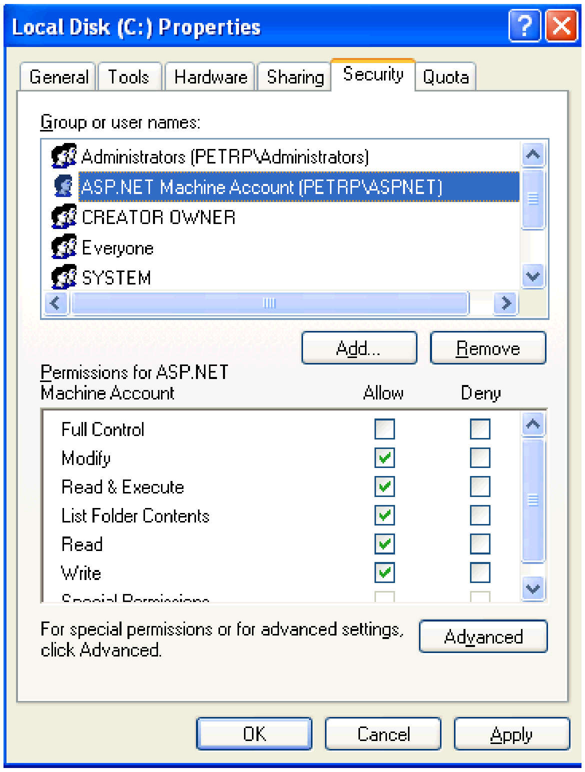
3. Choose your local computer in the pop-up window and click **OK**.



4. Enter **aspnet** into the box and click **Check Names**. The name should be resolved to **<your computer name>\ASPNET**. Click **OK**.

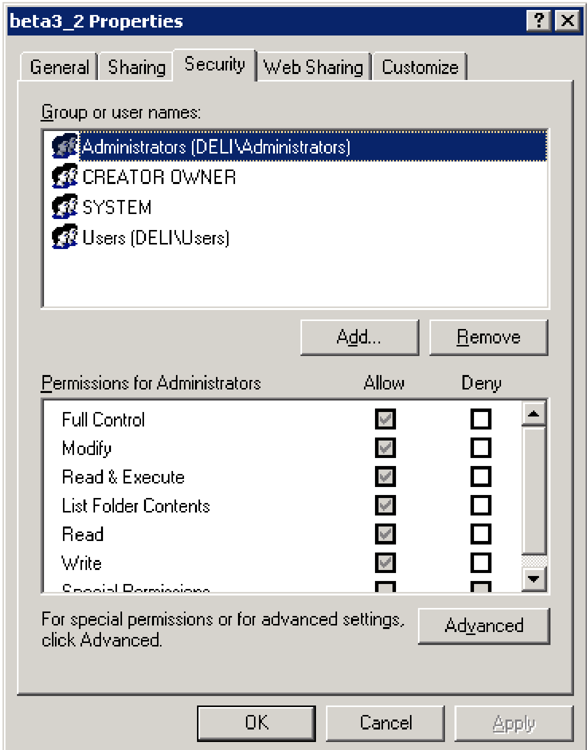


5. The account is added to the list of accounts. Grant the **Modify** permission to the account and click **OK**.

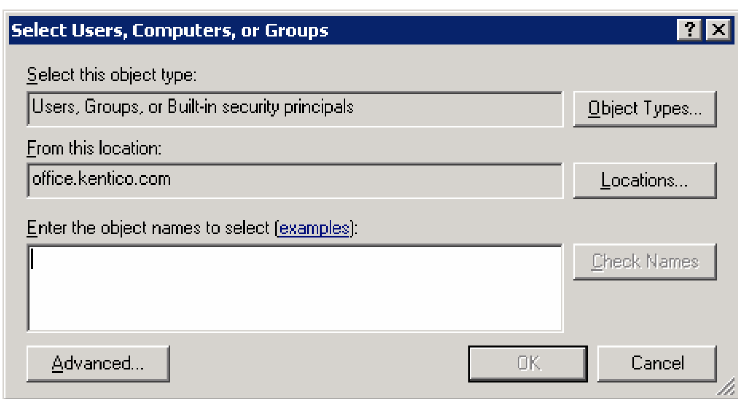


3.6.5.5 Solution on Windows 2003

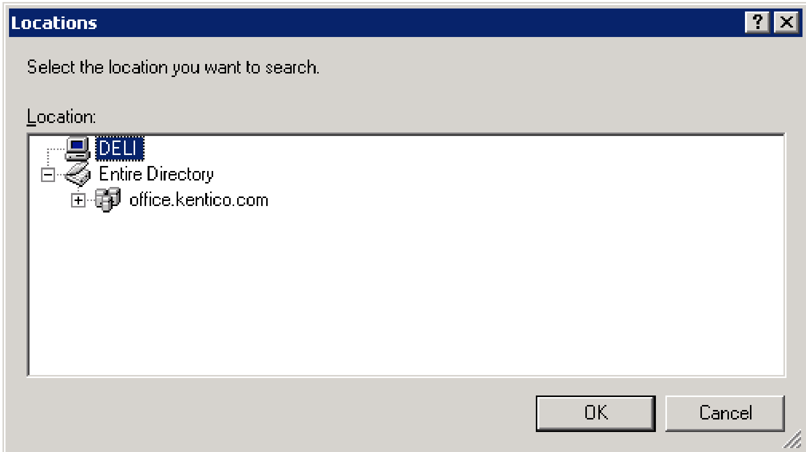
1. Open Windows Explorer, locate the folder with your website, right-click the folder and display its **Properties**. Choose the **Security** tab and click **Add...**.



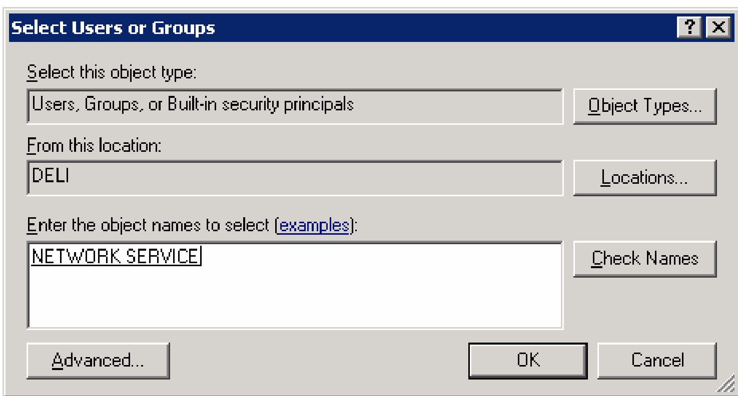
2. The **Select Users, Computers and Groups** dialog appears. Click **Locations...**.



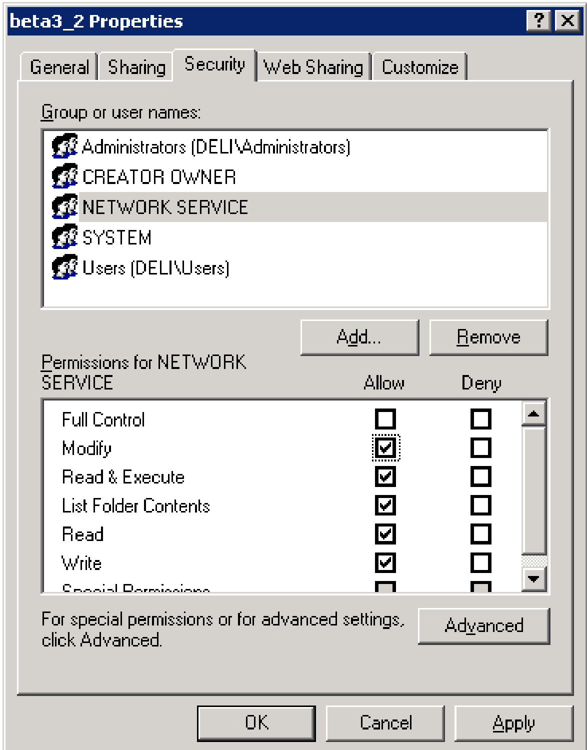
3. Choose your local computer and click **OK**.



4. Enter *network service* into the box and click **Check Names**. The name should be resolved to **NETWORK SERVICE**. Click **OK**.



5. The account is added to the list of accounts. Grant the **Modify** permission to the account and click **OK**.



3.7 System backup and recovery

Backup

To backup the data of Kentico CMS instance, you need to:

1. Backup the Kentico CMS database.
2. Backup the files of the Web application.

Recovery

To recover the system, you need to:

1. Recover the Kentico CMS database.
2. Recover the website files and create a virtual directory for it.

Part IV

Content management

4 Content management

4.1 Overview

The content can be managed through a browser-based, WYSIWYG user interface. You need to sign in to Kentico CMS Desk using your user name and password and you need to have appropriate permissions.

Links for access to the user interface

Kentico CMS user interface, i.e. **CMS Desk** and **Site Manager**, can be accessed via the following links, where the first part of the link is dependent on the installation and domain name. If the default URL of your website is *http://localhost/KenticoCMS*, then:

- The default URL of **CMS Desk** is *http://localhost/KenticoCMS**/CMSDesk***.
- The default URL of **Site Manager** is *http://localhost/KenticoCMS**/CMSSiteManager***.

Please note: The beginning of the addresses may be different depending on your installation and domain name, but the highlighted end of the URL is always in this format.

Default user name and password

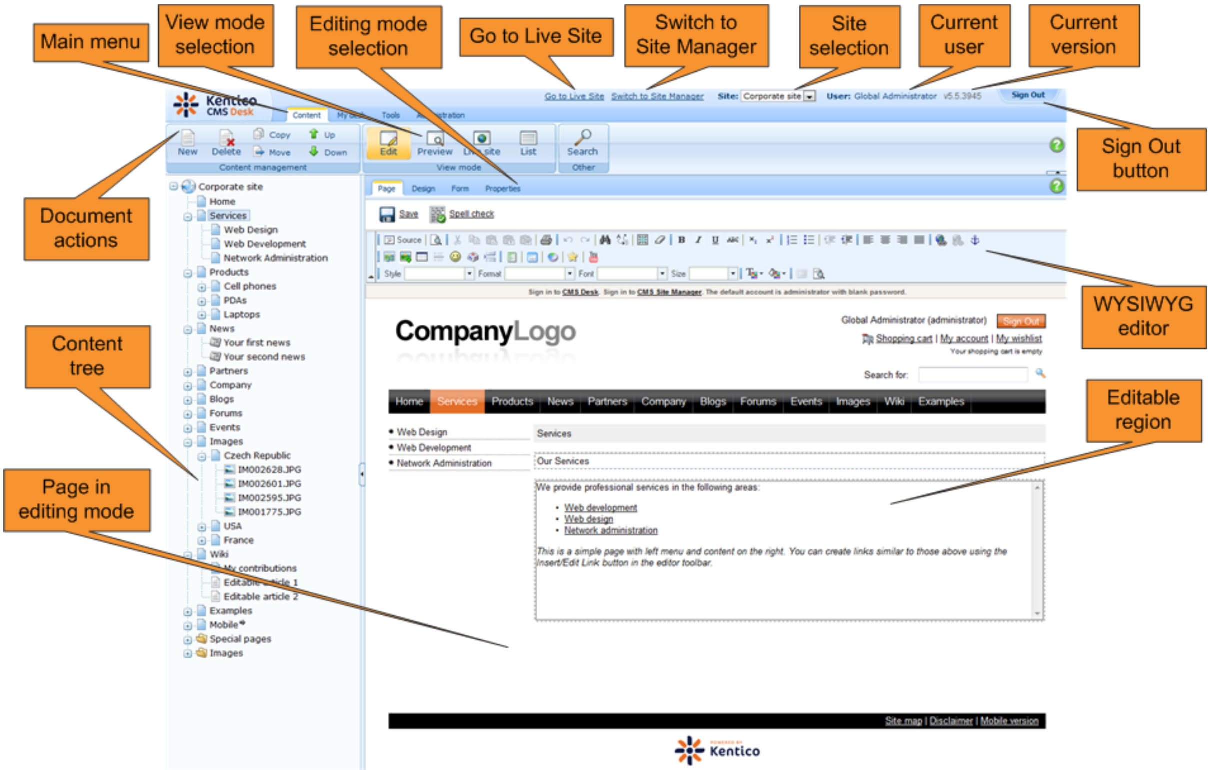
To log on to a Kentico CMS website for the first time, use the default logon credentials:

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

It's highly recommended that you change the password after you finish the installation.

User interface overview

The following figure shows the CMS Desk user interface.



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

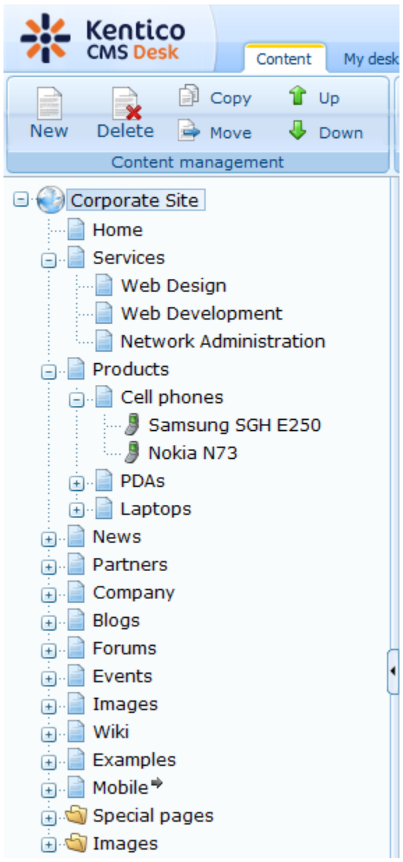
- **Main menu** with Content, My Desk, Tools and Administration tabs.
- **Content tree** that represents the site map of the website and allows you to organize the structure of documents and choose document that appears on the right side of the screen.
- **Document actions** toolbar with buttons for creating new documents, deleting, copying, moving and sorting documents.
- **View mode selection** - allows you to choose between editing, preview, live site view and list view.
- **Editing mode selection** - you can choose to edit page content, design the page template, edit the document fields, product properties or document properties.
- **Go to Live Site** - this action redirects you to the title page of the currently edited website, logged under the same user account that you used to log into the user interface. This is a more convenient way than using the **Sign out** button and logging in on the live site afterwards.
- **Switch to Site Manager** - redirects you to Site Manager, the other part of the system's user interface. This option is only available to global administrators.
- **Site selection** - this drop-down list is used to select the currently edited website. Only those websites that the current user can edit are available in the drop-down list.
- **Current user** - user name of the current user.
- **Current version** - version of Kentico CMS.
- **Sing Out button** - clicking this button log you off the user interface and redirects you to the title page of the live site. This button is only displayed if *Forms authentication* is used. When using *Windows authentication*, the link is not displayed.
- **WYSIWYG editor** - allows you to edit text stored in **Editable regions**, change text formatting or insert graphics into the text.
- **Page in editing mode** - this is where you can view and edit the document selected in the content tree, in the mode selected in the view mode and editing mode toolbar.

See also: Kentico CMS overview -> Where is the content stored?, Kentico CMS overview -> How do I

edit content?

4.2 Organizing pages, files and documents

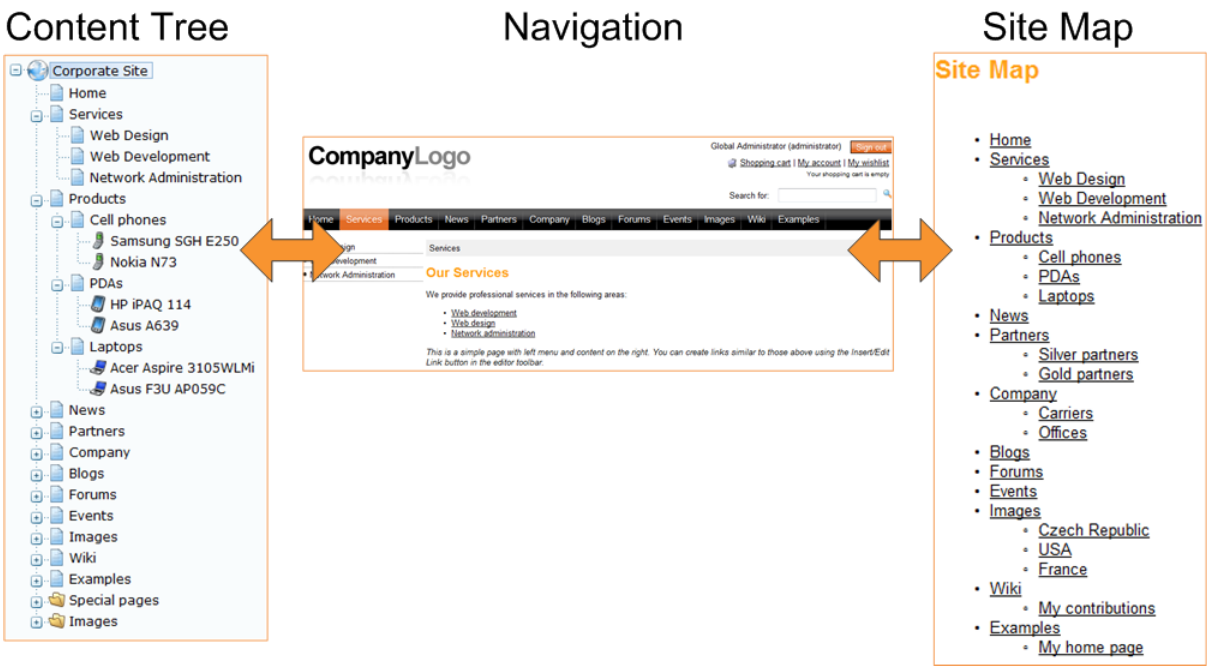
All content in Kentico CMS is stored in a **tree hierarchy**. You can see the content tree on the left in **CMS Desk -> Content**:



The tree hierarchy provides many advantages:

- It organizes the page in a logical structure that represents the (dynamic) site map.
- It ensures easy-to-navigate information architecture.
- It provides a logical categorization of pages and documents.
- The content of sub-pages can be nested inside the parent pages.
- The position of the document is reflected in its URL that consists of the document path in the tree hierarchy,
  such as `/products/lcd-displays/nec-52vm.aspx`.
- The structure allows you to define permissions for a particular site section and inherit them to underlying items.

The following figure shows how the content tree defines the navigation and site map of the website:



Document types

Each document is of some type - it can be a page, a news item, an article, a product specification, etc. Each document type has its own:

- fields (data structure)
- editing form layout
- transformations (design)
- queries

and other settings.

Document types are fully customizable - you can add, modify and delete custom fields. The advantage of using custom document types is that you can define custom structure of documents and store content (data) separated from design. This can be done in **Site Manager -> Development -> Document types**.

More details can be found in the Document types chapter.

Pages and documents

All items in the content tree are basically documents. However, there's a special type of documents called **pages**. The pages (such as /Home, /Products/PDAs) display the content and they are displayed as menu items by default (this can be also customized).

Unlike pages, structured documents (such as news item /News/Your first news in the sample Corporate Site) contain structured data that can be displayed on the pages.

While pages usually contain unstructured content in the form of editable regions that can be edited on the **Page** tab, the structured documents contain structured and typed data stored in document type-specific database tables and edited on the **Form** tab.

You will typically use **structured documents** when you need to display a **list of items**, such as list of news, list of products, etc.

Page versus Form

There are two types of content: content stored in editable regions on the page and content stored in forms. The following table compares both approaches:

	Editable regions on the page	Form
Content structure	Simple content structure, only text-based content.	Complex content structures, 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 providing truly WYSIWYG editing.	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 modifies using SQL queries or API.
Examples of use.	Home page, contact page. Generally: pages with simply structured or unstructured, text-only content. The editable regions are usually used only in connection with documents of type **Page (menu item)**.	News, product specification, event details, job opening, etc. Generally: pages with structured content where you need to separate content from design and keep the content in its original data type. The form-based content is usually used in connection with documents of type **News**, **Product, Article,** etc.

Organizing media files

There are two types of files you need to manage on the website:

- **website design files -** Images and Flash files that are used on the website design template, such as logo, background images, menu images, etc. These should be stored in the file system as a part of the application theme as explained in the App themes chapter.
- **Media files and document files** (images, Flash movies, Word documents, PDFs, etc.) that are published on the website and are part of the content editable by editors. These should be uploaded to the content tree as documents so that they can be managed by the content editors and so that you

can apply all content-related features (permissions, workflow, versioning, multilingual support) to files as well.

You can find more details on files in chapter File management.

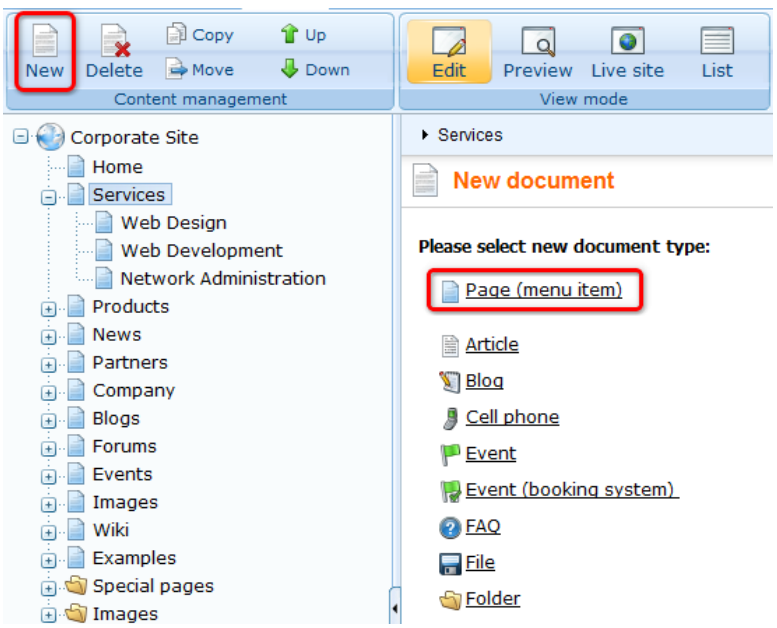
See also: Where is the content stored?

4.3 Editing content

4.3.1 Creating a new page

When you're creating a new page, you need to go through the following steps:

1. Click the **New** button in the main toolbar and choose to create a new document using the **Page (menu item)** document type.

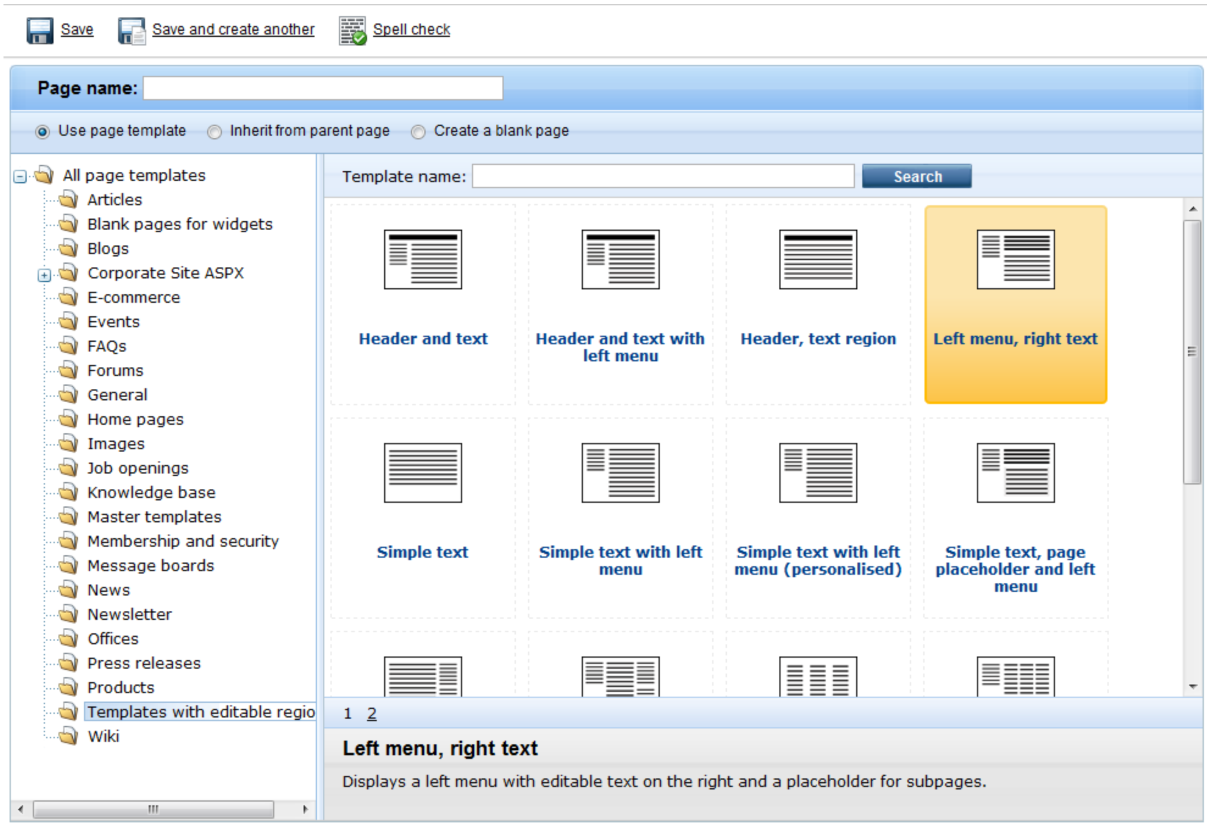


2. Enter the **page name** and select the **page template**. The page template defines the layout and design of the new page.

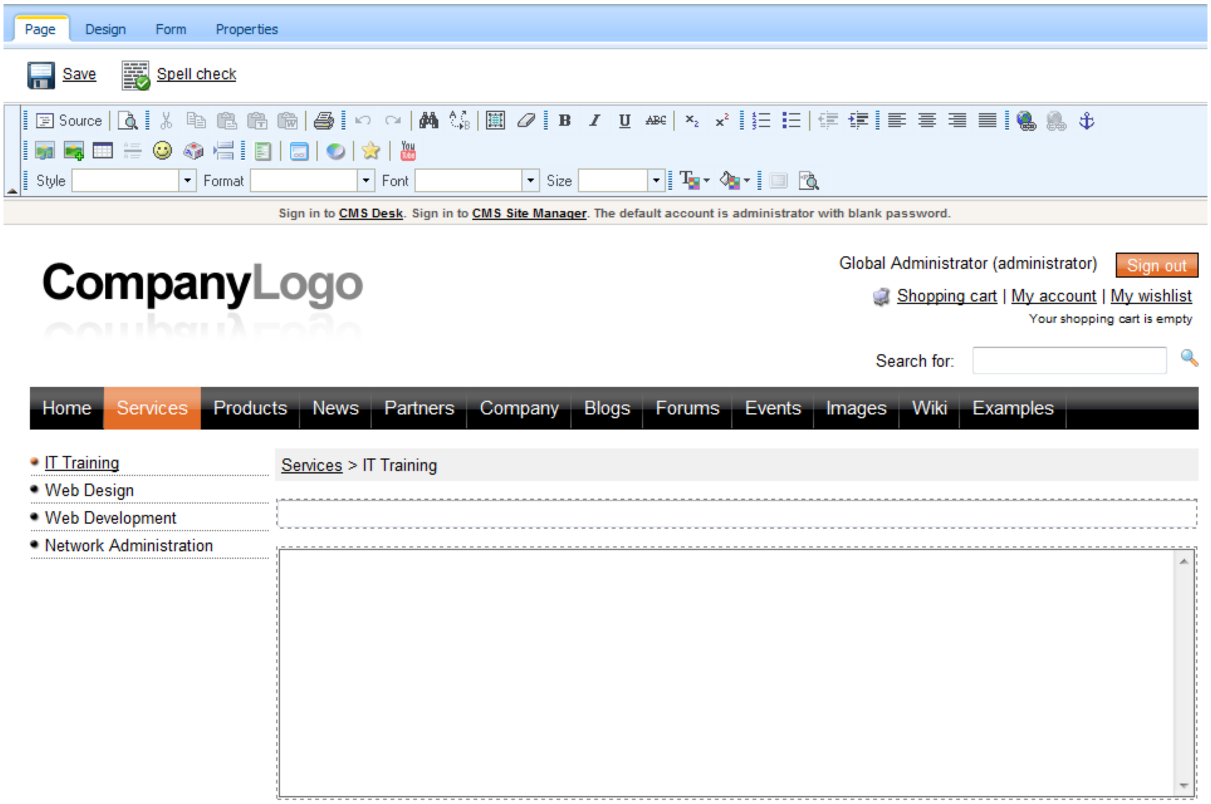
Alternatively, you can choose one of the following options when creating a new page:

- **Inherit from parent page** - use this option if you want the new page to use the same page template as the parent page.
- **Create a blank page using this layout** - this option is available only to site administrators/ developers who can use it to create a new page that doesn't use any existing page template (it's a page with an ad-hoc page template). You can learn more in Content tree and page templates.

Click **Save**.



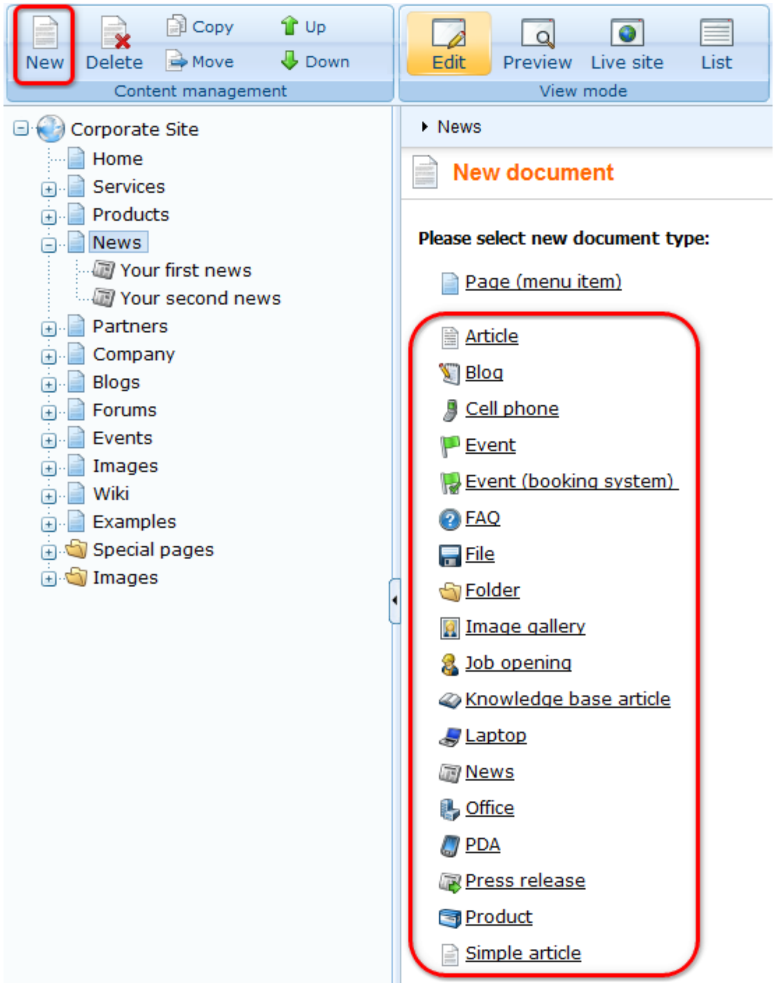
3. You're redirected to the **Page** tab and you can enter some content. Click **Save** to save your content changes.



4.3.2 Creating a new structured document

You can create new documents in **CMS Desk -> Content**.

1. Click the document under which the new item should be placed, click **New** and choose the required document type.



Available document types

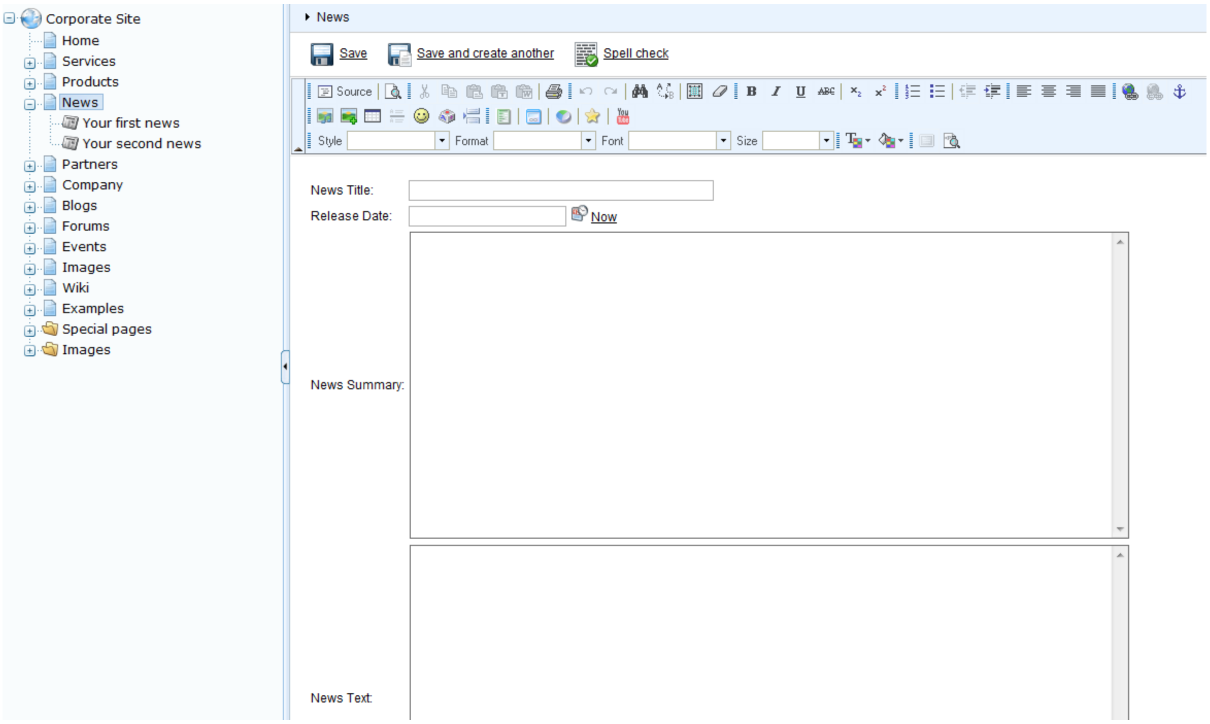
The types of documents that can be created under the selected document depend on the type of the selected document. If the required document type is not available, the site administrator needs to add it in Site Manager -> Development -> Document types -> ... edit parent document type ... -> Child types.

2. You are then redirected to the appropriate editing form. For example, if you chose to create a new **News** document in the previous step, you would get an editing form like in the screenshot below. Enter some sample values like the following:

- **News title:** My testing news
- **Released date:** 8/15/2007 (the release date displayed on the website)
- **News summary:** Some news summary.
- **News text:** Some news text.
- **Publish from:** <leave empty for now, it can be used to specify when the document goes live>
- **Publish to:** <leave empty for now, it can be used to specify when the document expires>

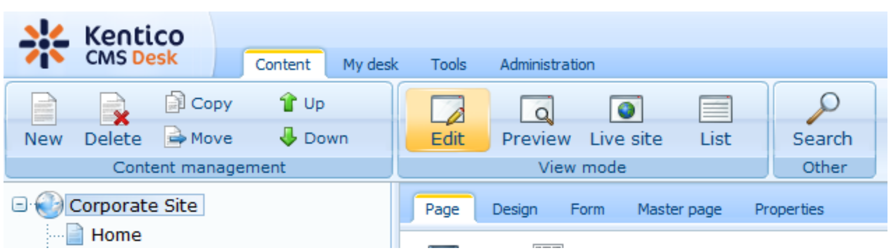
Now you can click **Save** to save the document and continue editing. You can also click **Save and**

create another to save the news document and create another news document in the same location. The later option is useful if you need to create several documents of the same type in the same location at once.



4.3.3 Previewing documents

When you open the page in CMS Desk -> Content section, the default mode is Edit:



The Edit mode allows you to edit the page content and properties.

You can also choose the following modes:

- **Preview** - this mode shows you the page as it will be published on the site, before it's published (if you're using workflow - see box below for details).
- **Live site** - this mode shows the page as it's currently displayed on the live site to site visitors.
- **List** - this mode shows the list of all documents under the currently selected document. It's useful if you have too many documents in a single site section and need to browse them effectively.

Preview mode and workflow

The preview mode works as expected **only if the displayed document uses workflow** (see Workflow overview for details). When you create a new document and want to preview it before you approve it in your step, click the document and click **Preview**. You will see the document as it will be displayed after it's published.

If you do not wish to use a real workflow, but you'd like to use the preview functionality, you can define a simple two-step workflow (Edit -> Publish) without any approval process. In this case, the editor can preview the content and publish it when it looks as required.

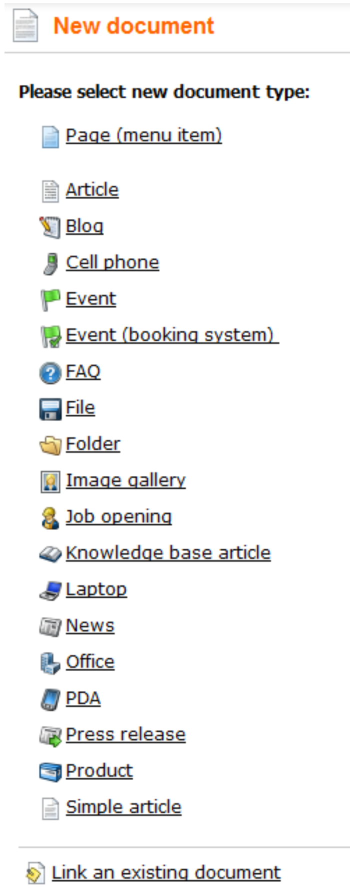
If you do not use workflow for the selected document, the **Preview mode** displays basically the same content as the **Live site** mode, the only difference may be caused by the fact that the preview mode doesn't use caching, so it may display also changes that are not visible on live site yet due to the caching.

4.3.4 Creating a linked document

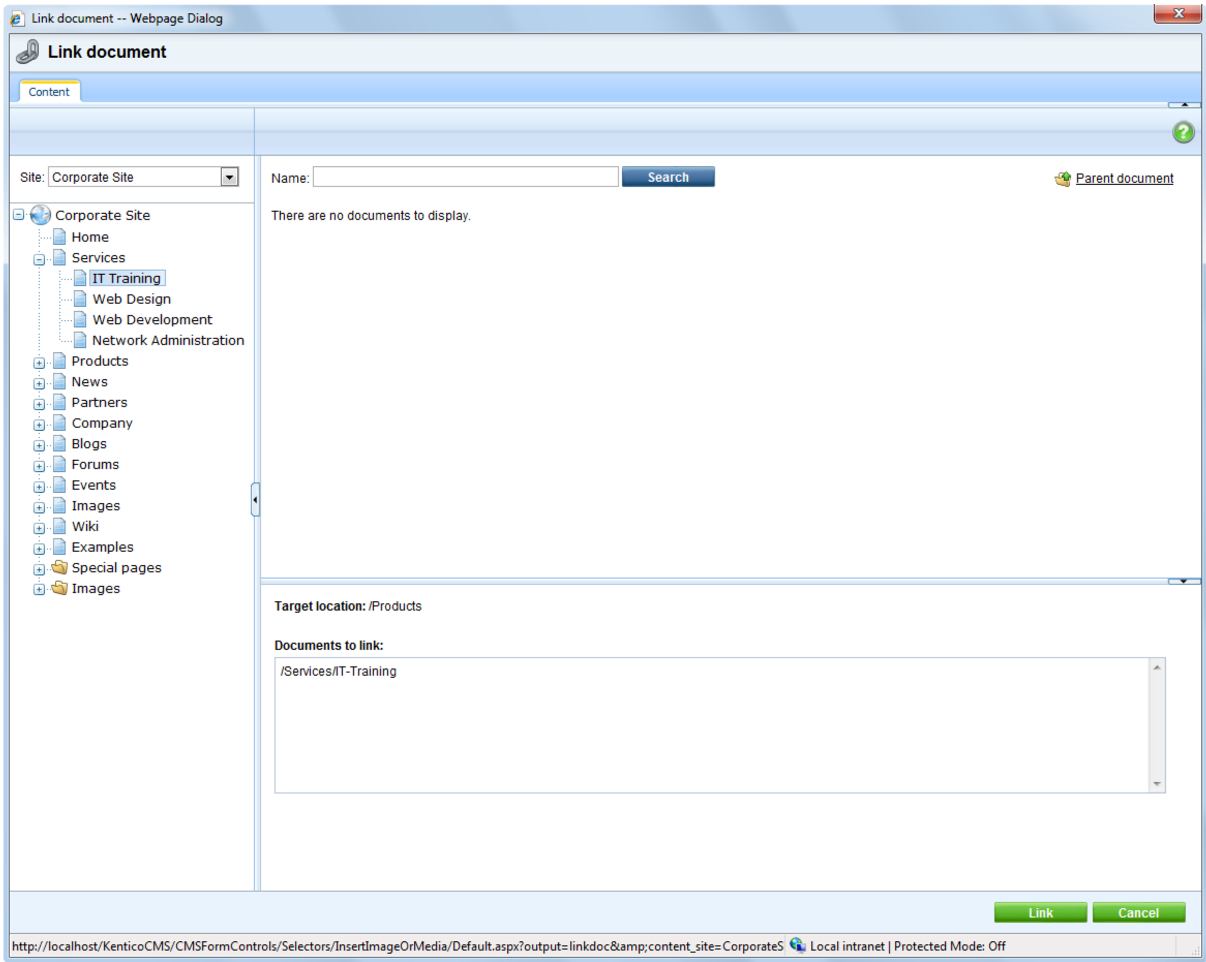
A linked document is a "stub" or "shortcut" of some existing document. It allows you to place a single document to multiple places in the content tree instead of creating its copies. Such a document is then displayed in the given part of the website, but when you edit it, you actually update the original document.

This feature is useful if you need to include a document (product) in multiple site sections (product categories), but keep only one instance of the document.

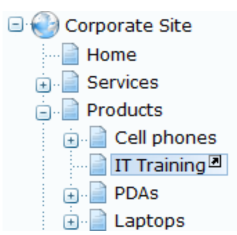
In order to create a linked document, click **New** and choose **Link an existing document** in the bottom of the screen:



Then, choose the document that should be linked to the current location:



Click **Save** to save the new link. The link to the document will appear with an arrow in the content tree:



You can see the list of all linked documents for the currently selected document in the Properties -> Linked docs dialog.

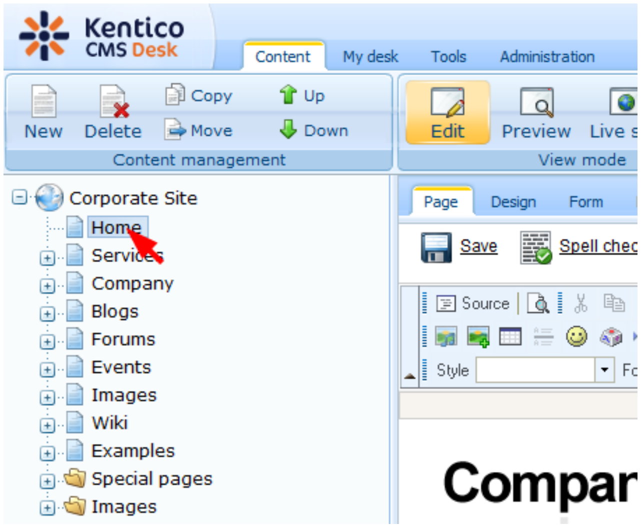
4.3.5 Drag-and-drop operations with documents

You can perform the following operations with documents quickly by dragging and dropping in the content tree.

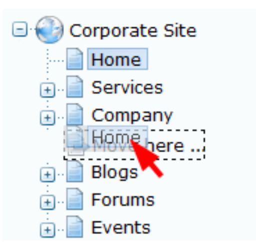
- **Move** - just drag and drop the document
- **Copy** - drag and drop while holding CTRL
- **Create linked document** - drag and drop while holding down CTRL+SHIFT

The following steps describe the process of dragging and dropping a document:

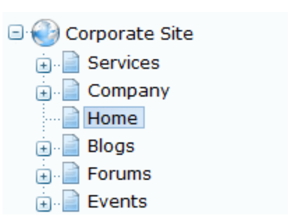
1. Select the document that you want to move.



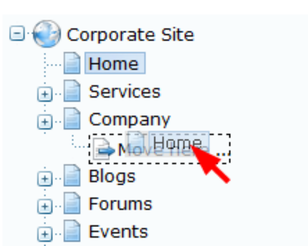
2. Move the mouse to the target location while still holding down the mouse button.



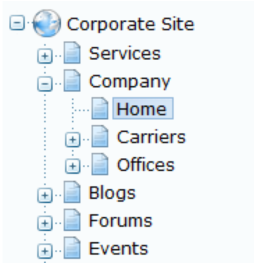
3. Release the mouse button. The document will be moved to the location where you dropped it.



4. If you drag the document a bit to the right ...



5. ... it will be placed under the document above it.



4.4 WYSIWYG editor

4.4.1 Overview

Kentico CMS comes with a built-in WYSIWYG editor. It is based on the FCKeditor, which is one of the best browser-based editors available on the market. However, it is possible to integrate your custom WYSIWYG editor instead of the built-in one.

Where is the editor available

You can come across the WYSIWYG editor in many parts of the system. However, its main use is related to the **Editable text** web part, which enables content editors to enter page content via the **Page** tab.

Another example of usage of the WYSIWYG editor is the **HTML area** form control that can be displayed e.g. on a document's **Form** tab, as a part of a BizForm or in the web part properties dialog of some web parts (e.g. **Static HTML** or **Silverlight application**).

WYSIWYG editor toolbar

The default toolbar looks like this:

As you can see, it provides functionality similar to Microsoft Word. Still, there are several buttons that may require additional explanation:

	**Paste as plain text** - this command pastes the content of your clipboard while cleaning out all formatting.
	**Paste from Word** - this command allows you to clean-up content pasted from Microsoft Word. It cleans up the HTML code so that it doesn't contain extra code and so that it fits your website design. You can find more details in this chapter.
	**Insert/Edit link** - creates a link from the selected text or inserts link into the text. See this chapter for more details.
	**Insert/Edit image or media** - inserts an image or other media into the text. See the Insert image or media chapter for more details.
	**Quickly insert image** - inserts an image from your local disk in a quick way, without any additional settings when inserting. See this chapter for more details.

	Insert BizForm - inserts an on-line form into the text. You can find more details on BizForms in the Modules -> BizForms chapter.
	Insert inline control - inserts an inline control into the text. You can find more details on inline controls in the Inline controls chapter.
	Insert poll - inserts a poll into the text. You can find more details on polls in the Modules -> Polls chapter.
	Insert/Edit YouTube video - inserts YouTube video. Further details can be learned in this chapter.

4.4.2 Insert image or media

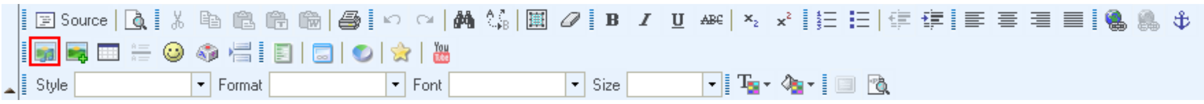
4.4.2.1 Overview

The Insert image dialog can be used to insert the following types of files:

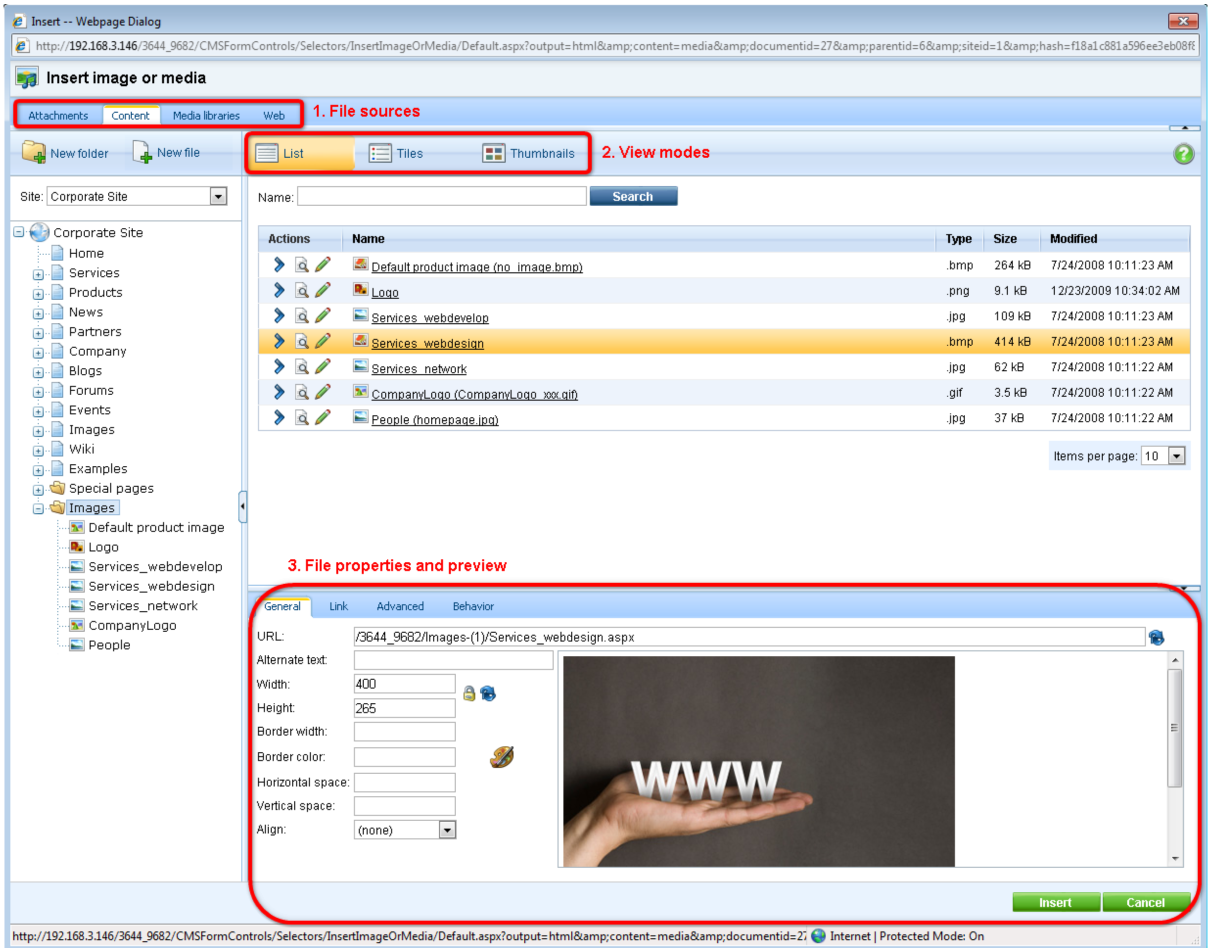
- **Images**: bmp, gif, ico, png, wmf, jpg, jpeg, tiff, tif
- **Audio**: wav, wma, mp2, mp3, mid, midi, mpga
- **Video**: avi, mp4, mpg, mpeg, wmv, qt, mov, rm
- **Flash**: swf

Custom types can be added as described in this topic.

The **Insert image or media** dialog can be opened by clicking the icon, as highlighted in the screenshot below.



After clicking the icon, the following dialog will be displayed:



The highlighted parts of the dialog have the following functions. Their position may vary based on the selected file source:

1. **File sources** - using these four tabs, you can select where the inserted files should be taken from; more info here
2. **View modes** - using these buttons, you can switch between different ways how listed files are displayed in the dialog; more info here
3. **File properties** - in this section, you can define properties of the file inserted into the text; the properties are different for images, audio/video and flash

General process of inserting image or media

1. Place the cursor in the appropriate position in the WYSIWYG editor.
2. Click the **Insert image or media** ( ) icon.
3. The dialog window opens. Select the appropriate file source tab according to where you want to add the file from.
4. Locate and select the file on the tab.
5. Appropriate properties according to the file type are displayed. Specify the required properties and click the **Insert** button.
6. The required code gets inserted into the WYSIWYG editor.

4.4.2.2 File sources

Using the four tabs at the top part of the dialog window, you can choose where the image or media should be inserted from. The following tabs are available:

Attachments

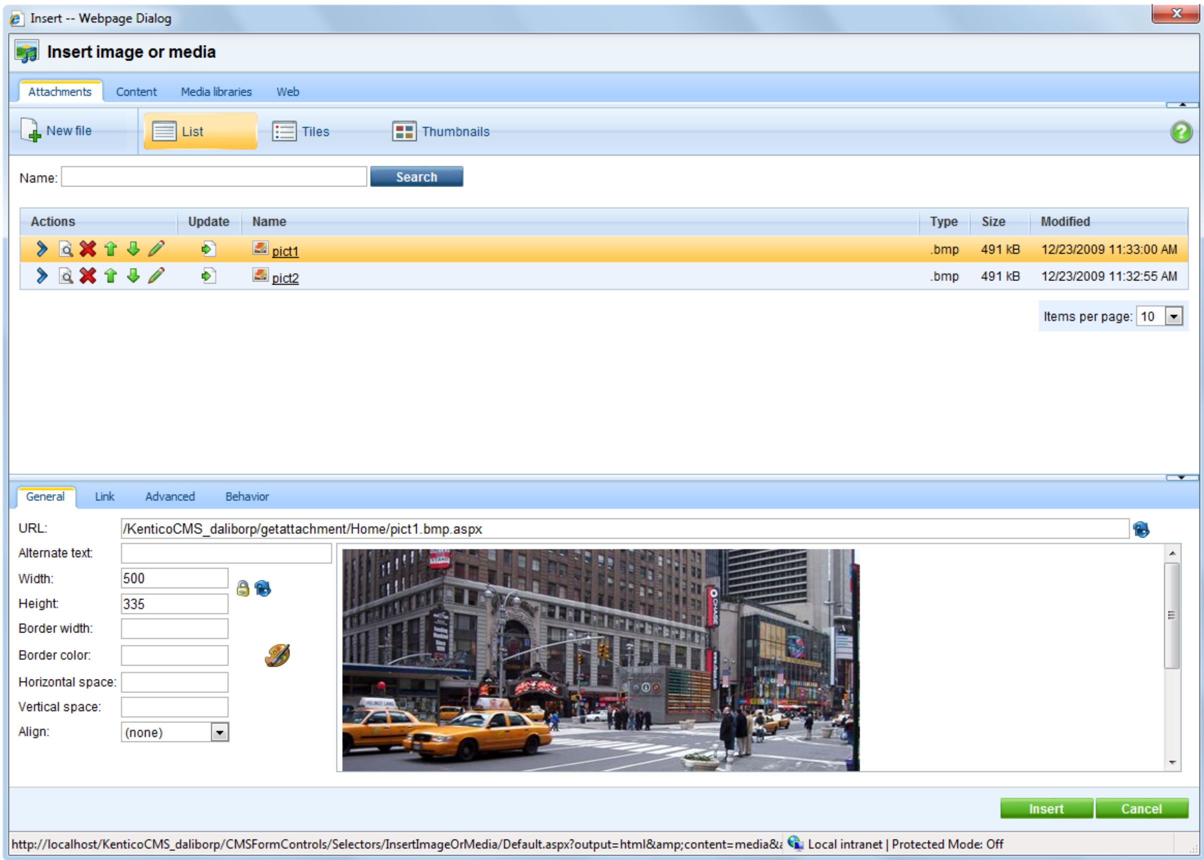
This tab is available only when the WYSIWYG editor is placed on some document (e.g. Form tab, user contributions, ...) so that the attachments of the current document can be offered. In other parts of the CMS, the tab is not displayed.

You can insert the document's unsorted attachments into the text from this tab. You can filter the listed attachments by **Name** using the filter above the list. For more information about document attachments, please refer to the Document attachments chapter of this guide.

You can select an attachment using the **Select** () icon or by simply clicking the appropriate line (or tile/thumnail in the other view modes).

You can upload new attachments to the document via this dialog using the **New file** () button. The following actions can be performed with the listed attachments:

- After clicking the **View** () icon, the attachment will be opened in a new window.
- Using the **Delete** () icon, you can remove the attachment from the document.
- Using the **Move up** () and **Move down** () icons, you can re-order the attachments. This option is available only in the **List** view mode. The order is stored in the **AttachmentOrder** property of each attachment. You can enter AttachmentOrder into the ORDER BY expression property of a displaying web part to have the attachments ordered accordingly.
- Images have also the **Edit** () icon available. Clicking this icon opens the image in the built-in image editor.
- Using the **Update** () icon, you can replace the original attachment with a new one.



Content

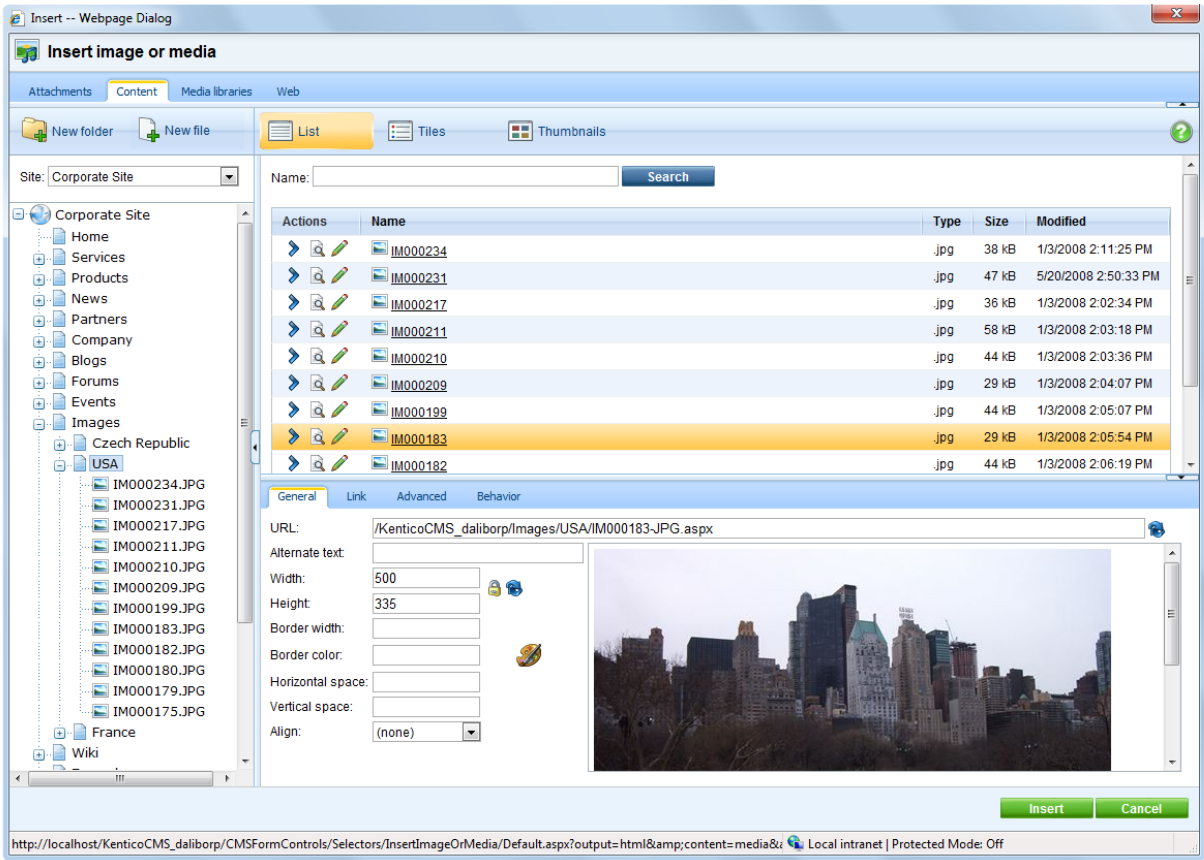
From this tab, you can insert files stored in the content tree of any of the sites running in the system. The site to insert from can be selected using the **Site** drop-down list, while its content tree will be displayed below. You can define which sites will be available and the starting alias path of the displayed content tree when defining the field in field editor, as described here.

If you select a node of the content tree, all insertable files stored under it will be listed in the main area. You can filter the listed files by **Name** using the filter above the list.

You can select a file using the **Select** ( ) icon or by simply clicking the appropriate line (or tile/ thumbnail in the other view modes).

New folders (cms.folder documents) can be created in the content tree via this dialog using the **New folder** ( ) button. You can also upload new files (cms.file documents) into the content tree using the **New file** ( ) button. You can also perform the following actions with the listed files:

- After clicking the **View** ( ) icon, the file will be opened in a new window.
- Images have also the **Edit** ( ) icon available. Clicking this icon opens the image in the built-in image editor.



Media libraries

From this tab, you can insert files stored in a media library. Depending on the settings described here, you can select a library using the set of three drop-down lists - **Site**, **Group** and **Library** - in the top right part of the dialog.

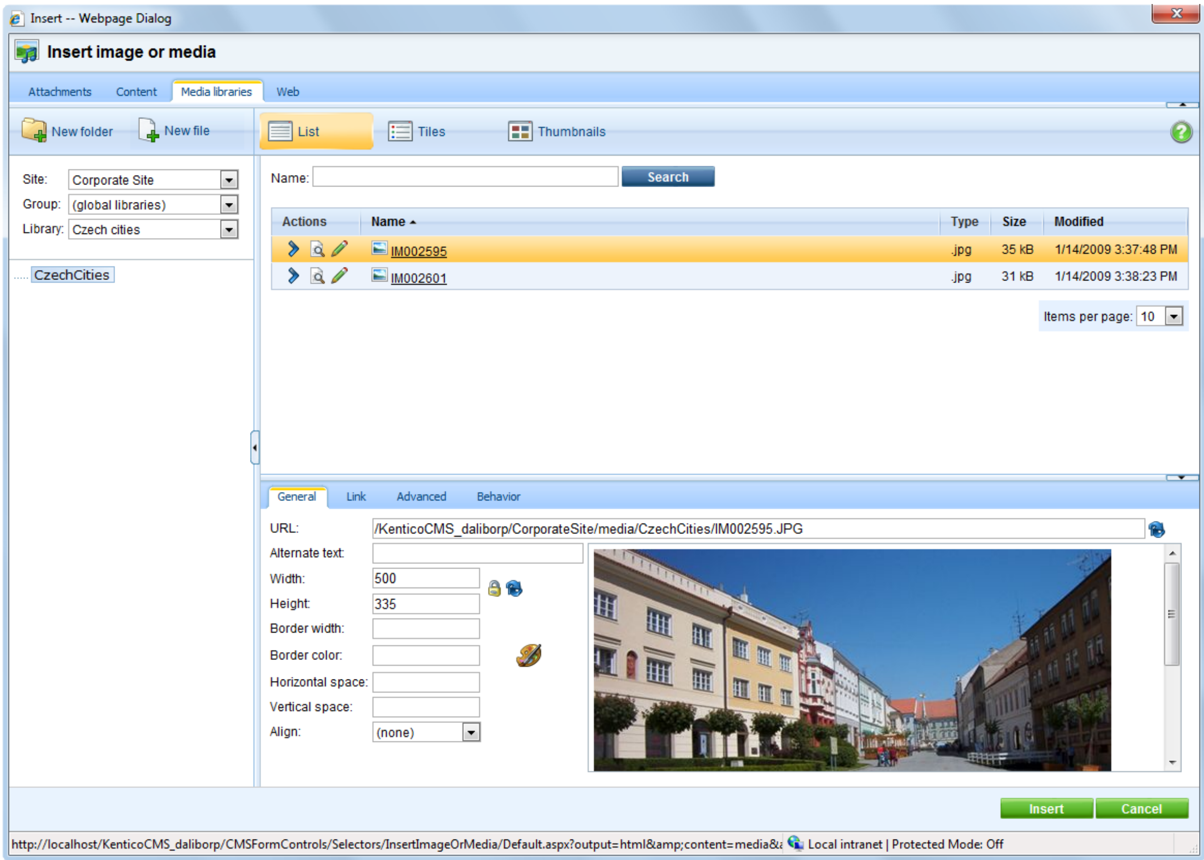
More information about Kentico CMS Media libraries can be found in this chapter.

When the library is selected, its folder structure is displayed below the drop-downs. After selecting a folder, its content will be offered in the main area. You can filter the listed files by **Name** using the filter above the list.

You can select a file using the **Select** ( ) icon or by simply clicking the appropriate line (or tile/ thumbnail in the other view modes).

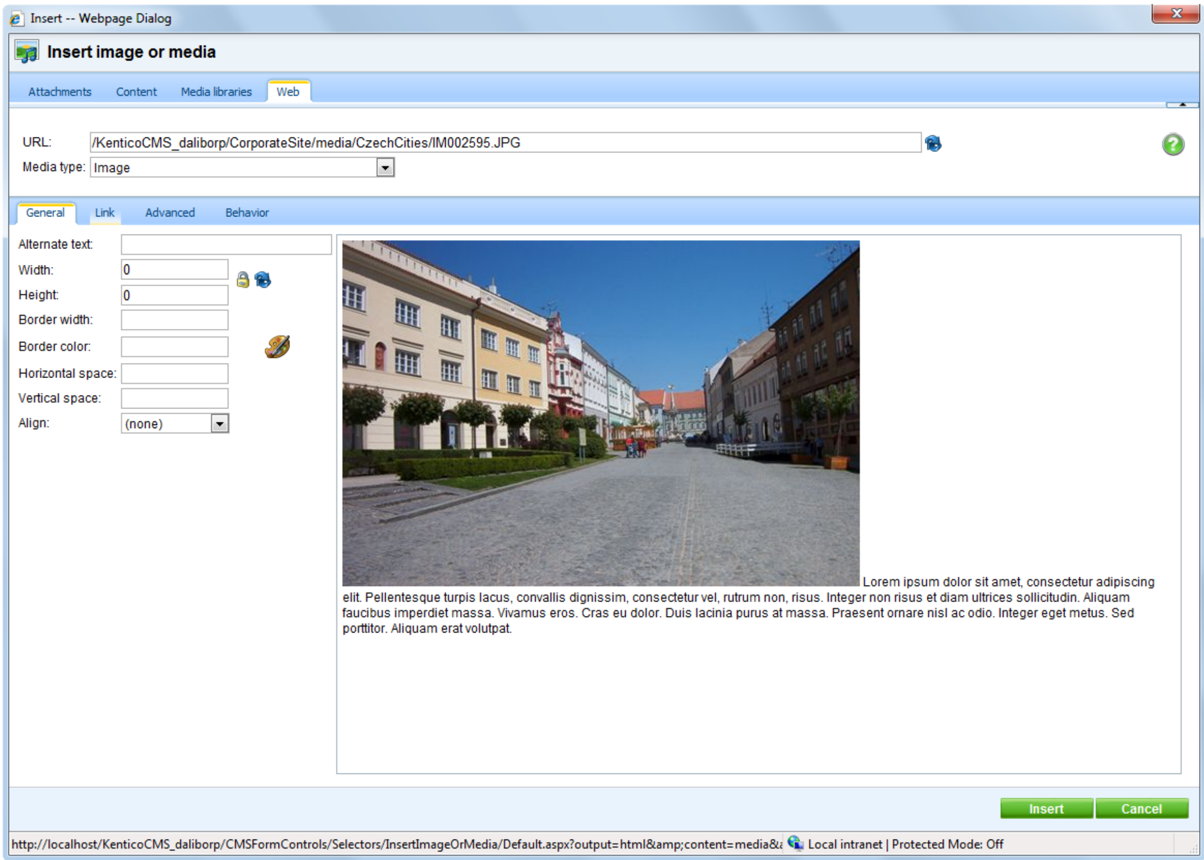
You can also perform the following actions with the listed files:

- After clicking the **View** ( ) icon, the file will be opened in a new window.
- Images have also the **Edit** ( ) icon available. Clicking this icon opens the image in the built-in image editor.



Web

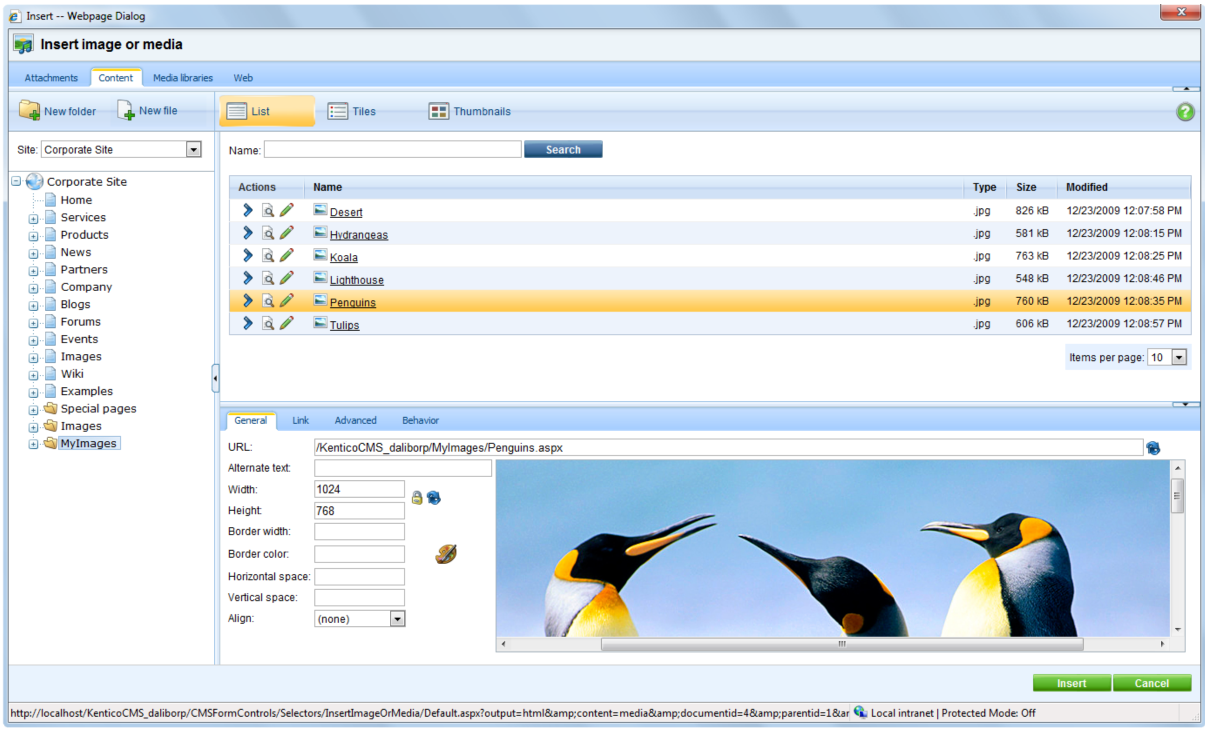
From this tab, you can enter an image, audio, video or flash from the Web by entering its URL. More information on how to use this tab can be found in this chapter.



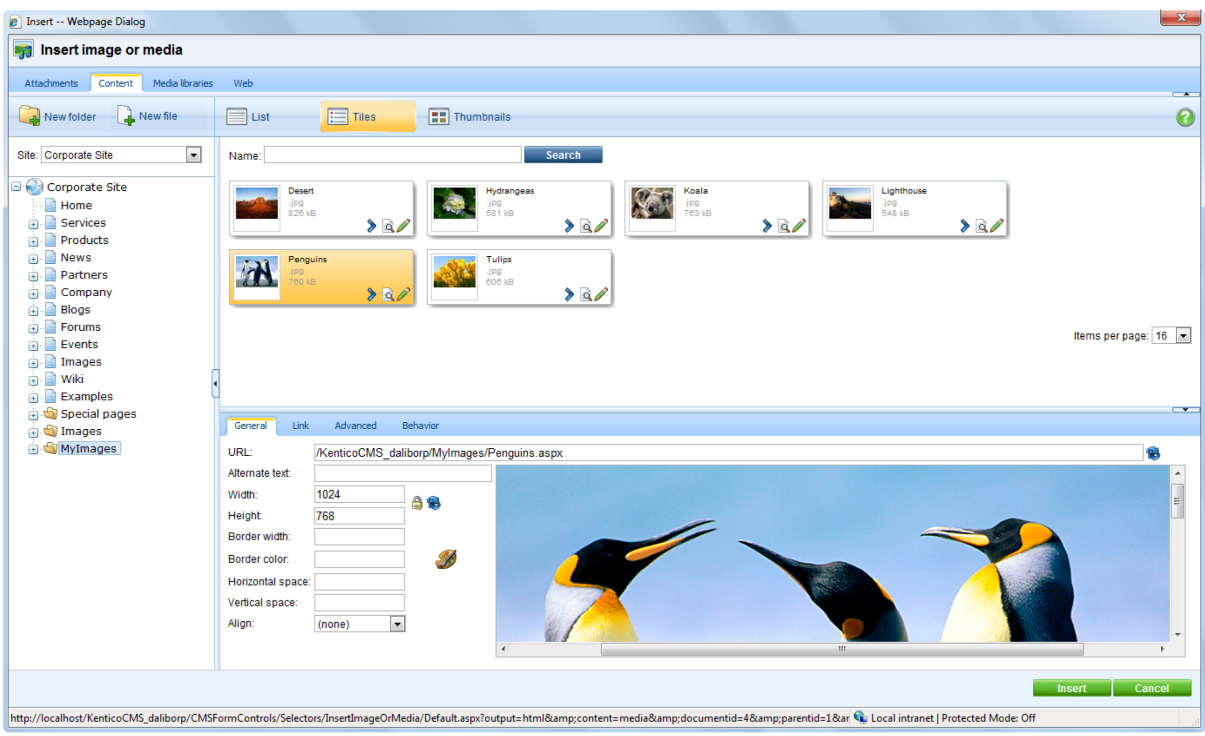
4.4.2.3 View modes

The following three view modes are available on the **Attachments**, **Content** and **Media libraries** tabs:

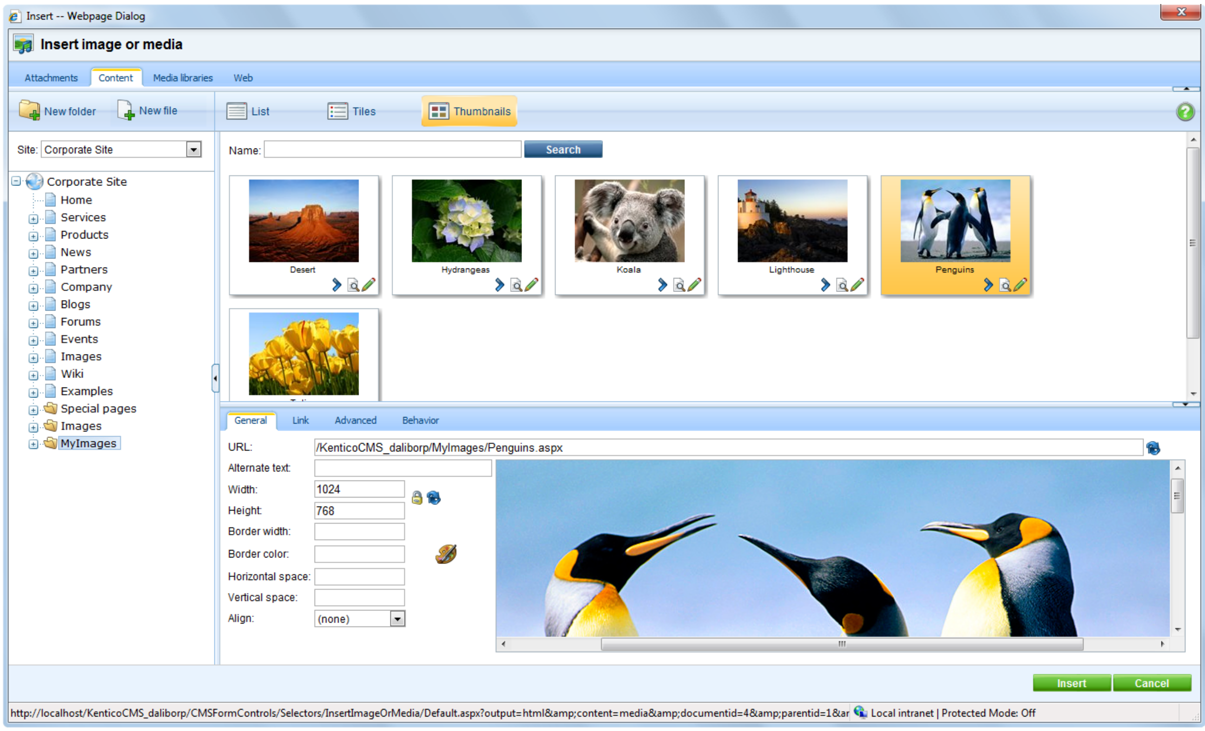
List



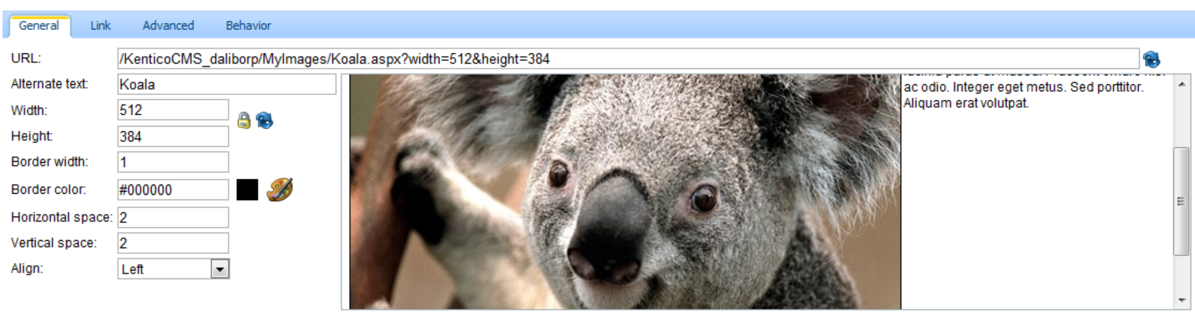
Tiles



Thumbnails



4.4.2.4 Inserting images



Inserting images via this dialog enables setting a number of properties of the image. If you want to insert an image from your disk in a quick way, without specifying any properties, you can use the **Quickly insert image** () action as described here.

When inserting an image, its properties can be set on the following four tabs:

General

- **Alternate text**: text displayed when the image is not loaded correctly
- **Width**, **Height**: width and height of the displayed image
  - aspect ratio can be locked (), which makes the second dimension recalculated automatically when you change one dimension, while the ratio between the two dimensions is kept
  - if it is unlocked (), dimensions can be entered manually into both fields, without the ratio being kept
  - you can also reload the default dimensions using the **Reset size** () icon
- **Border width**: width of the border around the displayed image

- **Border color**: color of the border around the image; has no effect when border width is not set
- **Horizontal space**: horizontal space between the image and the surrounding text
- **Vertical space**: vertical space between the image and the surrounding text
- **Align**: image alignment

Link tab

- **URL**: if set, the image will **become a link** to the resource defined by the entered URL. If set, **settings on the Behavior tab are overridden!**
- **Target**: destination where the linked resource should be displayed when the image is clicked

Advanced tab

- **ID**: identifier of the image HTML element
- **Tooltip**: text displayed when mouse cursor is placed over the image
- **Class**: image element CSS class
- **Style**: image element additional styles

Behavior tab

- **None**: image is inserted as standard image; when the image is clicked, no action is performed
- **Open full size in the same window**: image will become a link; when the image is clicked, its full size is displayed in the same window
- **Open full size in a new window**: image will become a link; when the image is clicked, its full size is displayed in a new window
- **Show larger size on mouse-over**: when the mouse cursor is placed over the image, the image is displayed in a "floating window" in the defined size; the image will be inserted as the **Image** inline control

6 ways how images can be inserted

1. Standard image

- **Behavior tab**: None
- **Link tab -> URL**: empty

The image is inserted as a standard image with no special behavior. No action is performed when the image is clicked or mouse-overed.

The output code looks like the following code sample:

```
<img style="width: 320px; height: 240px" alt="" src="/KenticoCMS41/MyImages/
Waterfall.aspx" />
```

2. Image with link

- **Behavior tab**: None
- **Link tab -> URL**: some URL is specified

The image functions as a link to the specified URL. When the image is clicked, user is redirected to the

URL in the same browser window.

The output code looks like the following code sample:

```
<a href="www.kentico.com"><img style="width: 320px; height: 240px" alt="" src="/
KenticoCMS41/MyImages/Waterfall.aspx" /></a>
```

3. Image with special behavior - full size in the same window

- **Behavior tab**: Open full size in the same window
- **Link tab -> URL**: empty

When the image is clicked, it is displayed in full size in the same browser window.

The output code looks like the following code sample:

```
<a target="_self" href="/KenticoCMS41/MyImages/Waterfall.aspx"><img style="width:
320px; height: 240px" alt="" src="/KenticoCMS41/MyImages/Waterfall.aspx" /></a>
```

4. Image with special behavior - full size in a new window

- **Behavior tab**: Open full size in a new window
- **Link tab -> URL**: empty

When the image is clicked, it is displayed in full size in a new browser window.

The output code looks like the following code sample:

```
<a target="_blank" href="/KenticoCMS41/MyImages/Waterfall.aspx"><img style="width:
320px; height: 240px" alt="" src="/KenticoCMS41/MyImages/Waterfall.aspx" /></a>
```

5. Image with special behavior - larger size on mouse-over

- **Behavior tab**: Show larger size on mouse-over; Height = xx; Width = yy
- **Link tab -> URL**: empty

When the image is mouse-overed, it is displayed in a new layer in a size defined on the Behavior tab as shown in the screenshot below.

In this case, the image is inserted as the **Image** inline control. The output code looks like the following code sample:

```
{^Image|(behavior)hover|(url)~/MyImages/Waterfall.aspx?width=200&amp;height=150|
(width)200|(height)150|(mouseoverwidth)400|(mouseoverheight)300^}
```

6. Image with special behavior - larger size on mouse-over with link

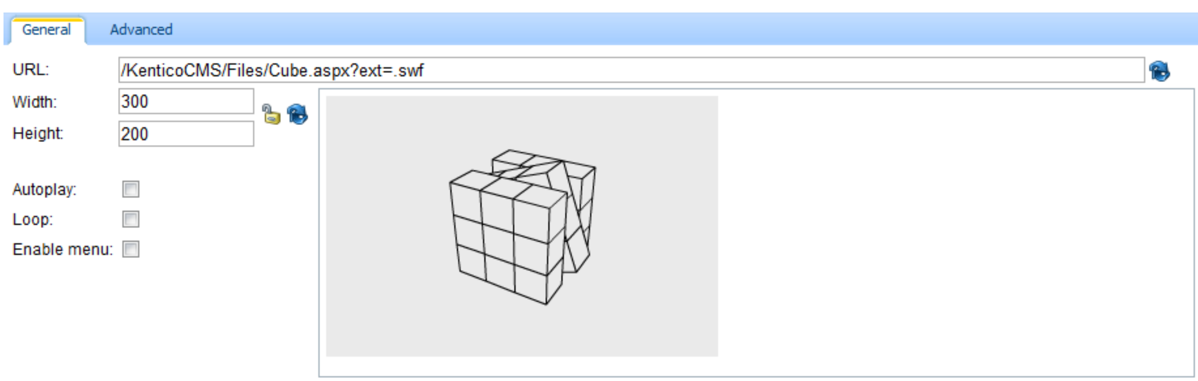
- **Behavior tab**: Show larger size on mouse-over; Height = xx; Width = yy
- **Link tab -> URL**: some URL is specified

When the image is mouse-overed, it is displayed in a new layer in a size defined on the Behavior tab as shown in the screenshot above. Furthermore, the image is clickable and when clicked, the user gets redirected to the specified URL in the same browser window.

In this case, the image is inserted as the **Image** inline control. The output code looks like the following code sample:

```
<a href="www.kentico.com">{^Image|(behavior)hover|(url)~/MyImages/Waterfall.aspx?
width=200&amp;height=150|(width)200|(height)150|(mouseoverwidth)400|
(mouseoverheight)300^}</a>
```

4.4.2.5 Inserting flash



When inserting flash, its properties can be specified on the following two tabs:

General

- **Width**, **Height**: width and height of the flash player; 300x200px is used by default
  - aspect ratio can be locked (), which makes the second dimension recalculated automatically when you change one dimension, while the ratio between the two dimensions is kept
  - if it is unlocked (), dimensions can be entered manually into both fields, without the ratio being kept
  - you can also reload the default dimensions using the **Reset size** () icon
- **Autoplay**: indicates if the video will play automatically when the player loads
- **Loop**: indicates if the player plays the video repeatedly in a loop
- **Enables menu**: indicates if flash options are available in flash context menu; flash context menu is displayed on right click of the flash player

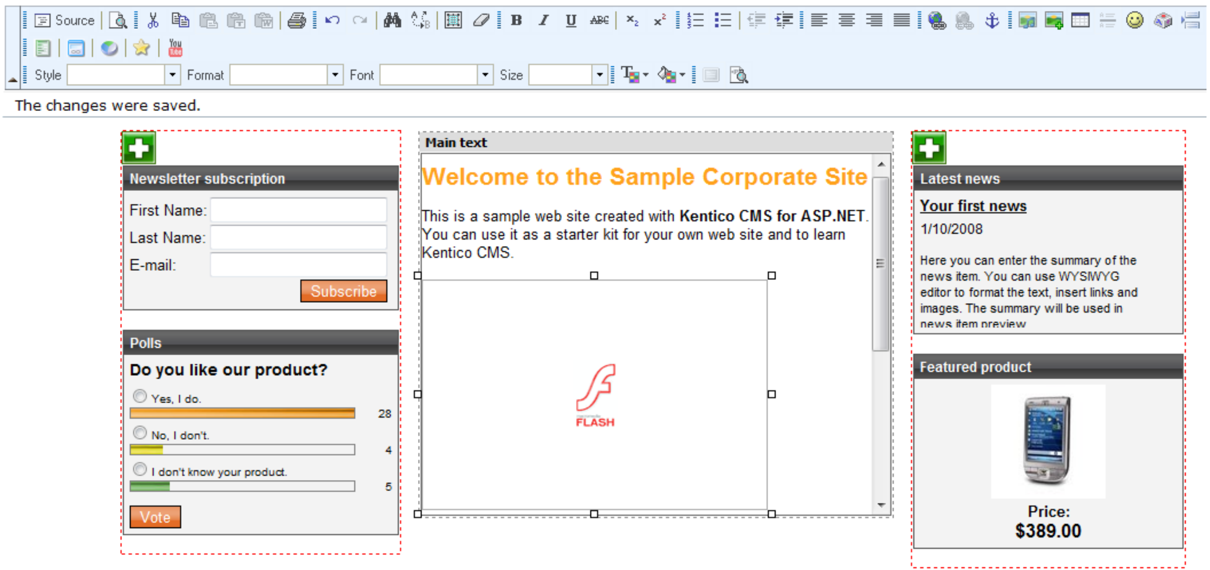
Advanced

- **Scale**: defines how the flash player stretches, shrinks or resizes when the browser window is resized
- **ID**: identifier of the flash HTML object
- **Advisory title**: text displayed when mouse cursor is placed over the flash player
- **Class**: flash element CSS class
- **Style**: flash element additional styles

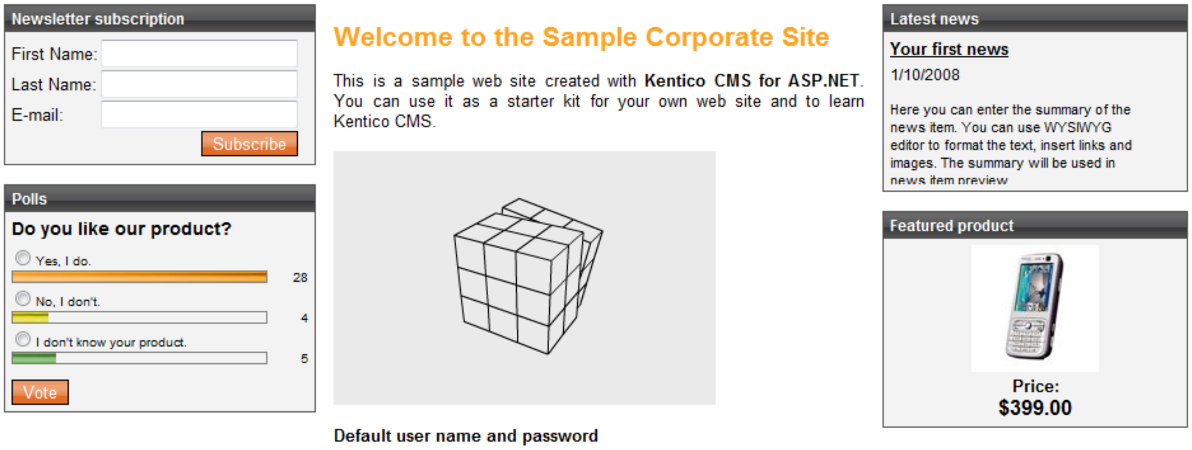
Flash is inserted into the output code as the **Media** inline control. The following code sample shows what the output code looks like:

```
{^Media|(type)swf|(url)http://127.0.0.1/KenticoCMS41/Files/Cube.aspx?ext=.swf|
(width)300|(height)200^}
```

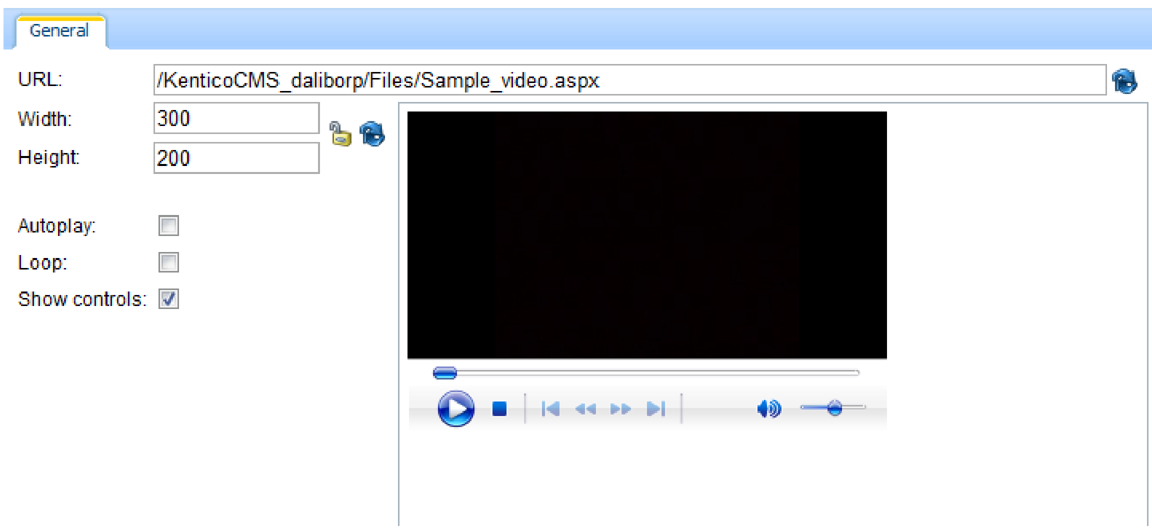
In the WYSIWYG editor, the flash is displayed only in the form of a box with the Flash logo, giving information about the size of the player:



And this is the result on the live site:



4.4.2.6 Inserting audio/video



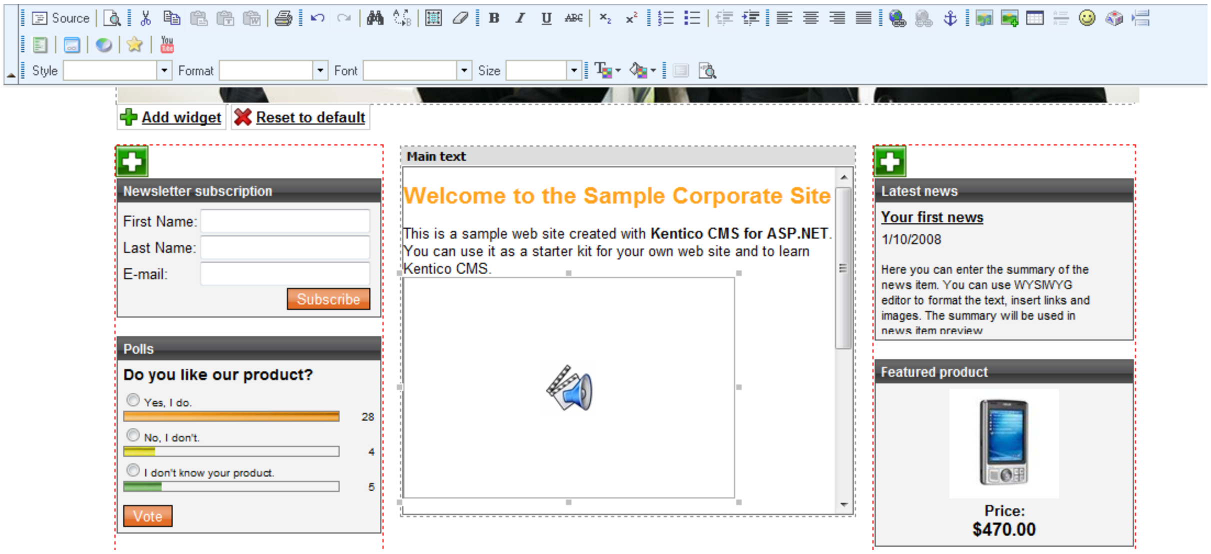
When inserting both audio or video, the following properties can be set:

- **Width**, **Height**: width and height of the media player; 300x200px is used by default
  - aspect ratio can be locked (), which makes the second dimension recalculated automatically when you change one dimension, while the ratio between the two dimensions is kept
  - if it is unlocked (), dimensions can be entered manually into both fields, without the ratio being kept
  - you can also reload the default dimensions using the **Reset size** () icon
- **Autoplay**: indicates if playback starts automatically when the page is loaded
- **Loop**: indicates if playback is performed repeatedly in a loop
- **Show controls**: indicates if playback controls (play, stop, fast forward, ...) are displayed; in some browsers, the controls may not be displayed if the player size is too small even when this option is enabled

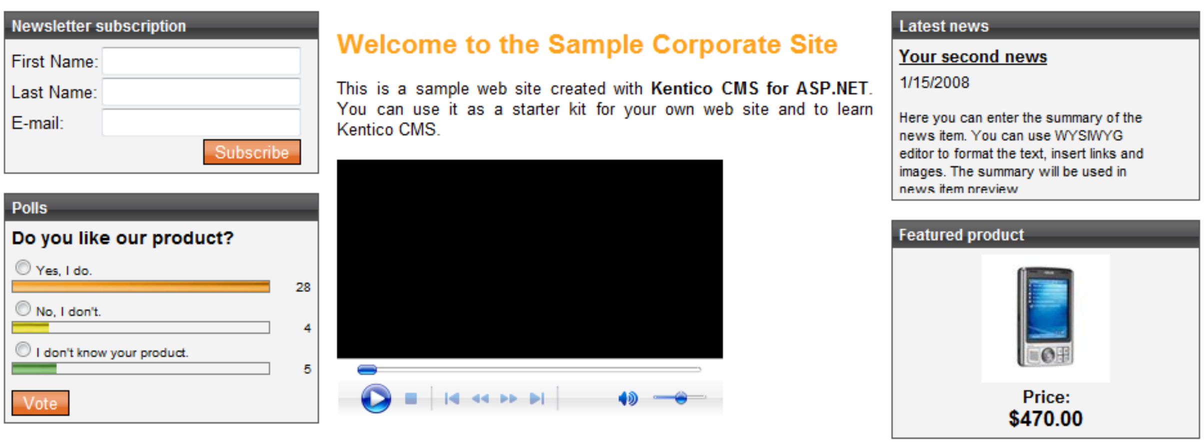
Audio and video is inserted to the output code as the **Media** inline control. The output code looks like the following code sample:

```
{^Media|(type)avi|(url)http://127.0.0.1/KenticoCMS_0713/Files/Sample_video.aspx|
(width)300|(height)200|(controls)1^}
```

In the WYSIWYG editor, the player is not displayed. Instead, only a box with audio/video icon is displayed, giving information about the size of the player on the live site:



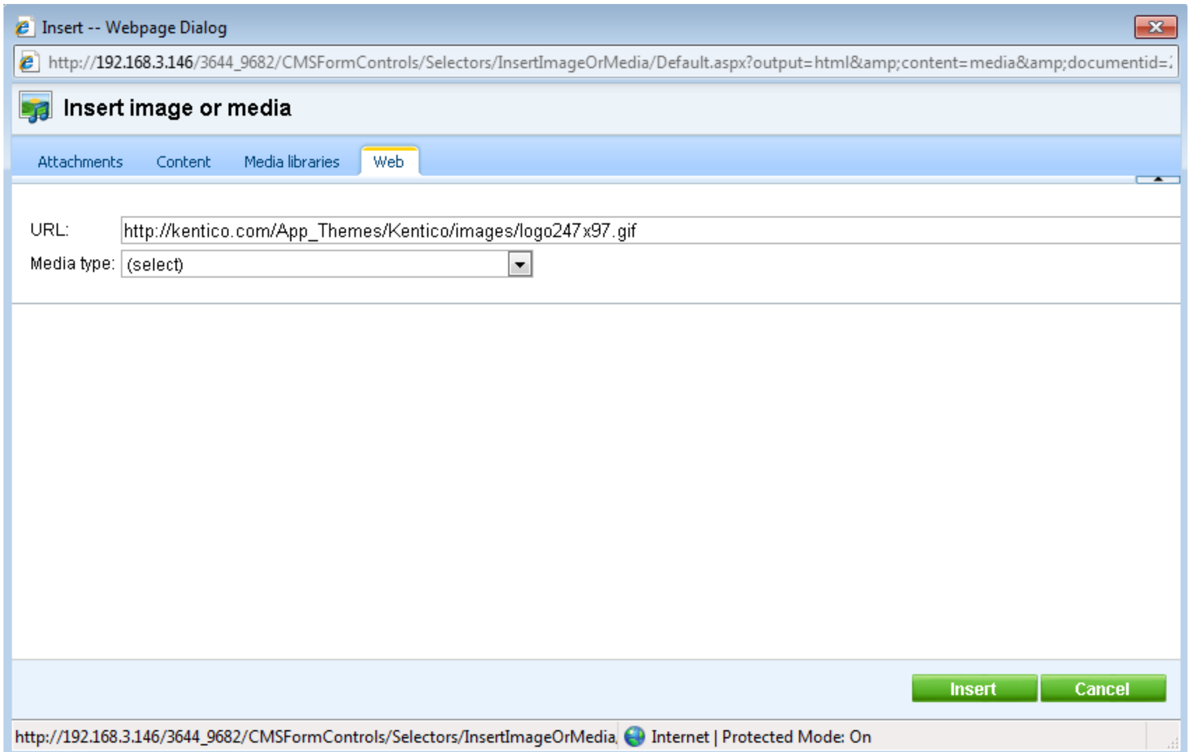
And this is how it looks on the live site:



4.4.2.7 Inserting image or media from Web

Via the Web tab, you can insert any of the file types enumerated in this chapter from web just by entering their URL. The generated code depends on the type of linked media and looks as the code samples mentioned in the previous three chapters.

The dialog initially looks as the following on the Web tab:



The general process of adding images or media from web is the following:

1. Enter the URL of the resource into the **URL** field.

2. Try automatic file type detection using the **Refresh** ( ) icon. In case that the file type can not be detected, you can still choose it manually from the **Media type** drop-down list.

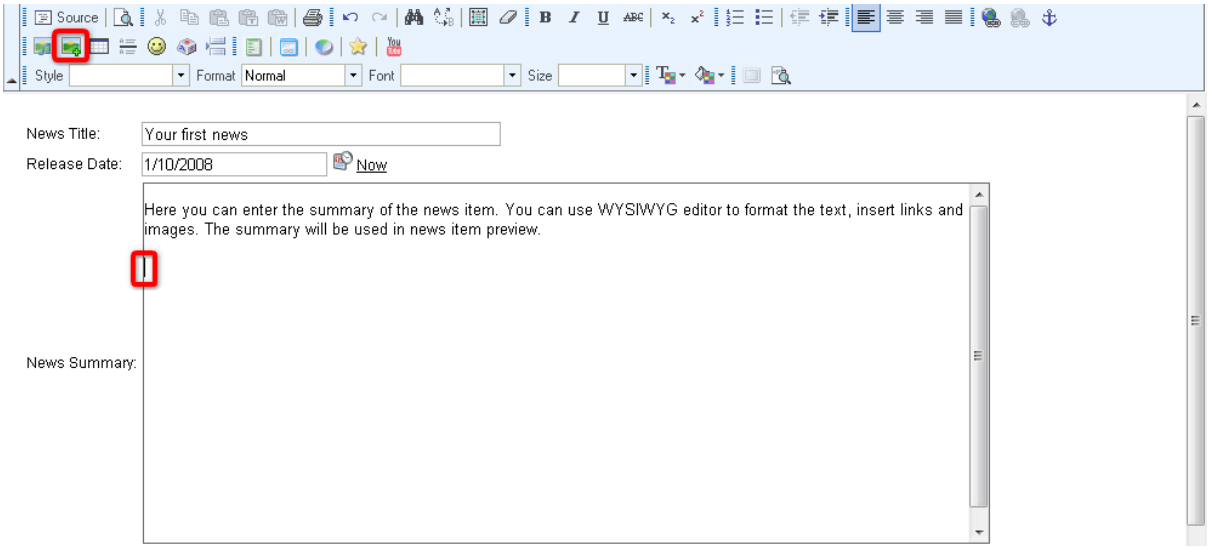
3. Based on the file type, properties will be loaded into the main area. The properties for particular file types are described in the Inserting images, Inserting flash and Inserting audio/video chapters.

4. Enter the properties and click the **Insert** button. The image or media file is inserted into the text.

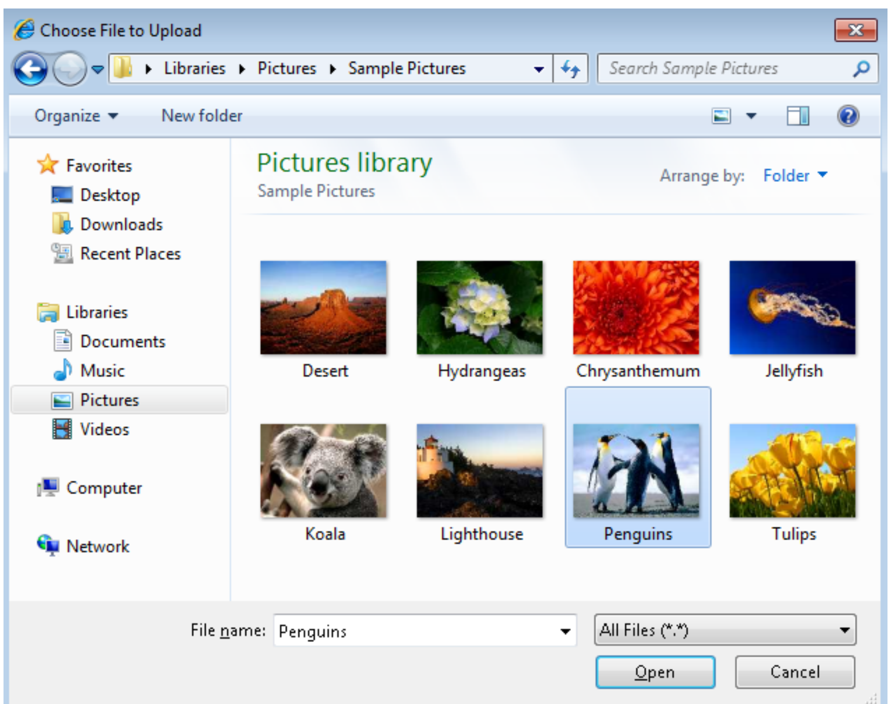
4.4.3 Quickly insert image

The Quickly insert image action can be used to insert an image from disk in a quick way. It takes just the following three steps to insert an image this quick way:

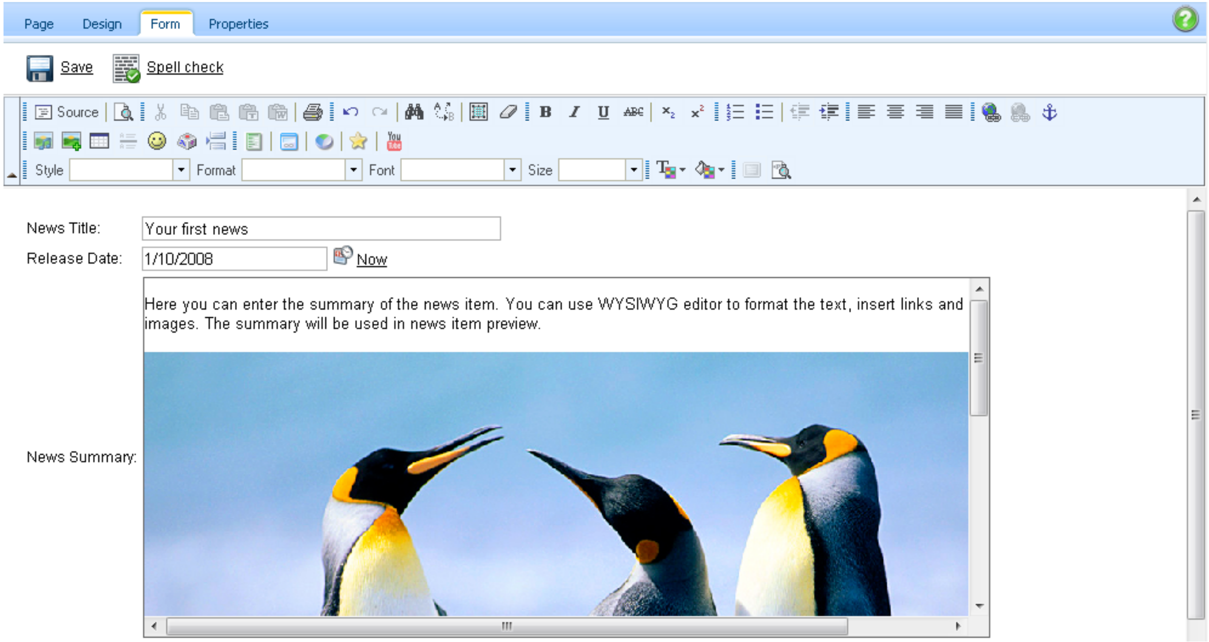
1. Place the cursor in the appropriate position and click the **Quickly insert image** ( ) icon on the WYSIWYG editor toolbar.



2. Your browser's **Choose file** dialog opens. Locate the file on your disk and click **Open**.



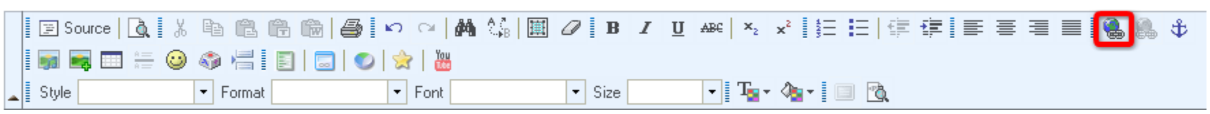
3. The image gets inserted into the text. At the same time, it is uploaded to the document as its attachment. No properties are specified during the insertion process, so the image is inserted in full size with no special behavior. However, you can now edit the image just as images inserted via the **Insert image or media** dialog. For more information, please refer to the Editing inserted items chapter.



4.4.4 Insert link

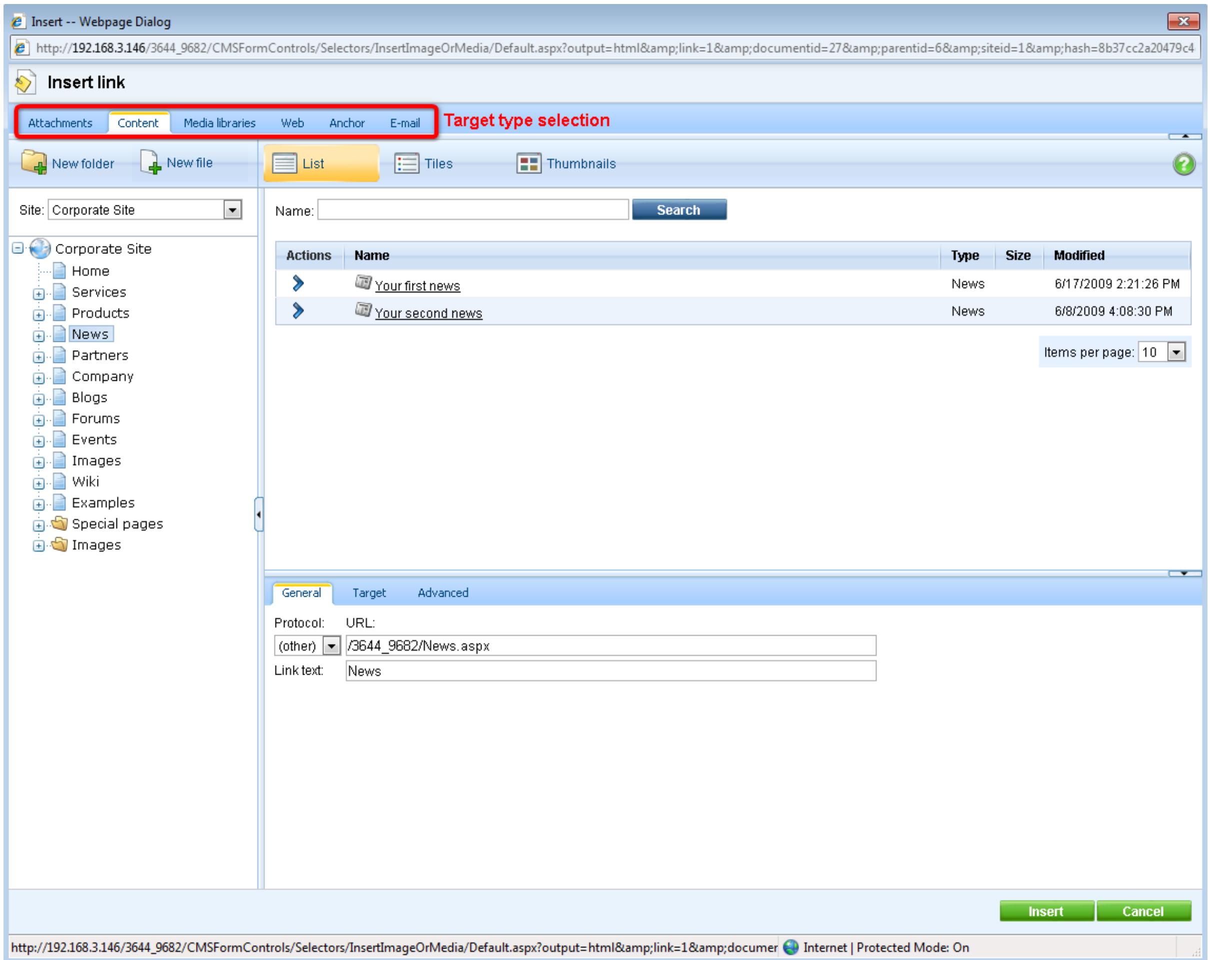
4.4.4.1 Overview

The Insert link dialog is accessible by clicking the **Insert link** ( ) icon in the WYSIWYG editor toolbar:



All links are inserted using the **<a>** tags and the following types of targets can be linked:

- Content within the CMS - via the **Attachments**, **Content** and **Media libraries** tabs
- Content anywhere on the Web - via the **Web** tab
- Anchors in documents - via the **Anchor** tab
- Mailto links - via the **E-mail** tab



General process of inserting a link

1. Select the text that should become a link or place the cursor to the position where the link text should be inserted.
2. Click the **Insert link** ( ) icon on the WYSIWYG editor toolbar.
3. Choose the appropriate tab and specify where the link should be leading.
4. Specify link properties and click the **Insert** button.
5. The link gets inserted in the text.

4.4.4.2 Link properties

On the **Attachments**, **Content**, **Media libraries** and **Web** tabs, you can see the following section for settings link properties:

You can specify the following properties on the particular tabs:

General

- **Protocol + URL**: address of the linked resource
- **Link text**: text of the link that will appear in the text; this field is visible only when inserting link into an empty space in the text area, i.e. when no text or object is selected

Please note: On the Web tab, only the last two tabs are displayed and the general properties are displayed above them.

Target

- **Target**: using this drop-down list, you can define where will the link be opened
- **Frame name**: this option is displayed only when the Target property is set to (frame) and specified the name of the frame where the target should be displayed

Advanced

- **ID**: identifier of the link HTML element
- **Name**: name of the link HTML object
- **Tooltip**: text displayed when the mouse cursor is placed over the link
- **Class**: link element CSS class
- **Style**: additional link element styles

4.4.4.3 Links to content within the CMS

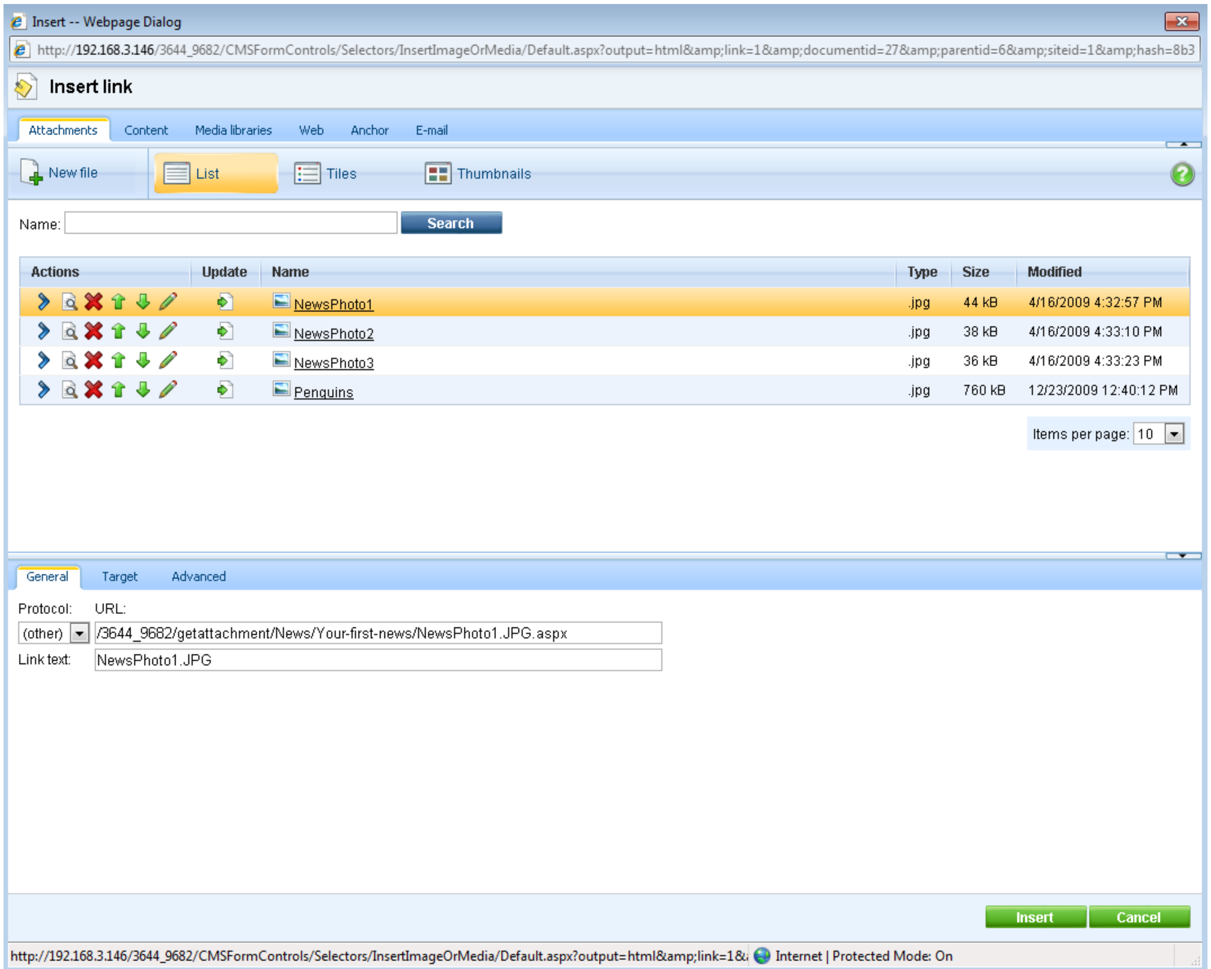
Links to content within the CMS can be inserted via the **Attachments**, **Content** and **Media libraries** tabs. Detailed descriptions of all possible actions that can be performed on the tabs can be found on this page of the **Inserting image or media** section.

Attachments

Via this tab, you can insert links to attachments of the current document. For more information about document attachments, please refer to the Document attachments chapter of this guide.

The following code sample shows what the output code looks like:

```
<a href="/3644_9682/getattachment/News/Your-first-news/NewsPhoto1.JPG.aspx">
NewsPhoto1.JPG</a>
```

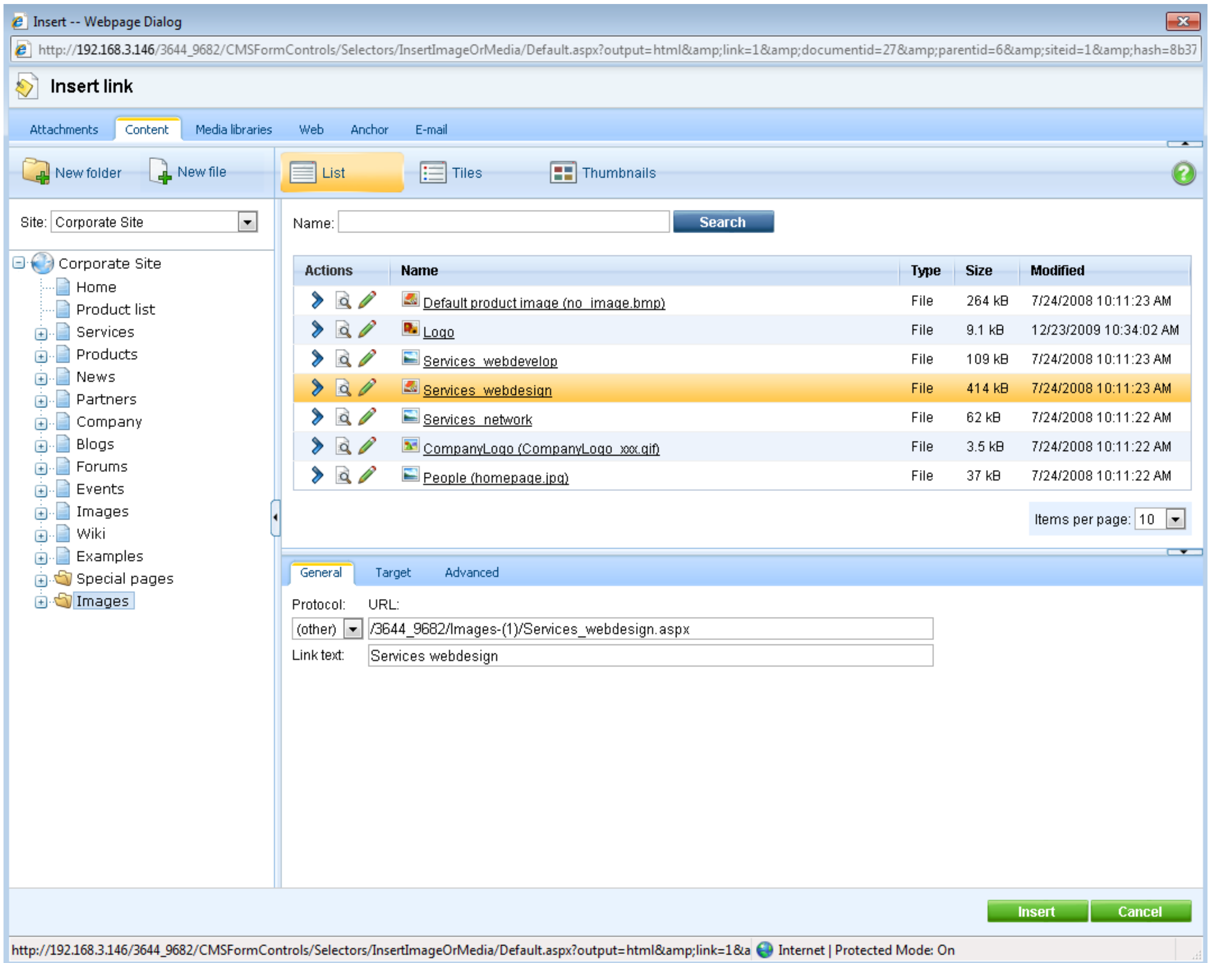


Content

Via this tab, you can insert links to any pages, documents or files within the content tree of a site. The site can be selected using the **Site** drop-down list, while its content tree will be displayed below. You can define which sites will be available and the starting alias path of the displayed content tree when defining the field in field editor, as described here.

The following code sample shows what the output code looks like:

```
<a href="/3644_9682/Images-(1)/Services_webdesign.aspx">Services webdesign</a>
```

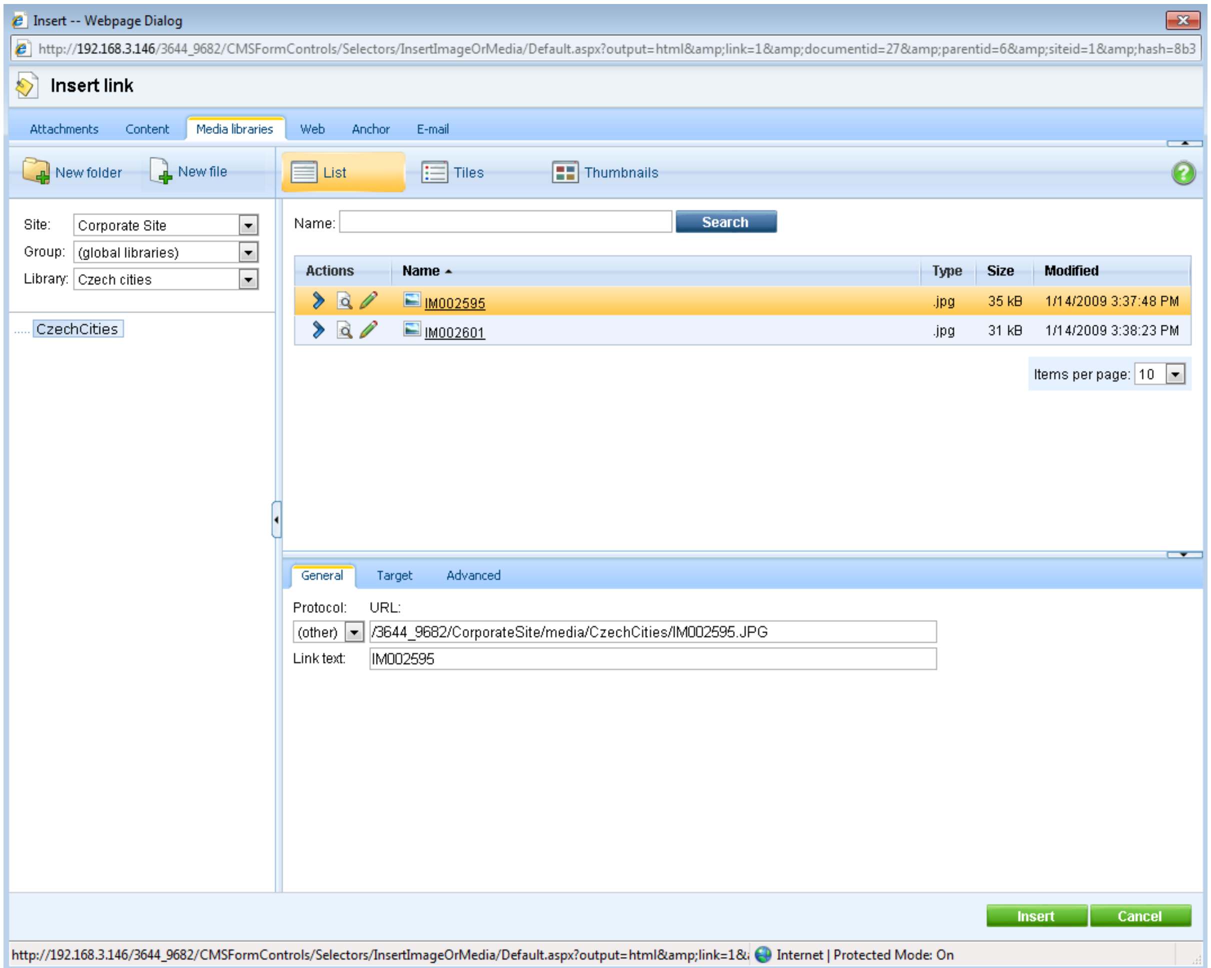


Media libraries

Via this tab, you can insert links to files stored within media libraries. Depending on the settings described here, you can select the library using the set of three drop-down lists - **Site**, **Group** and **Library** - in the top right part of the dialog.

The following code sample shows what the output code looks like:

```
<a href="/3644_9682/CorporateSite/media/CzechCities/IM002595.JPG">IM002595</a>
```

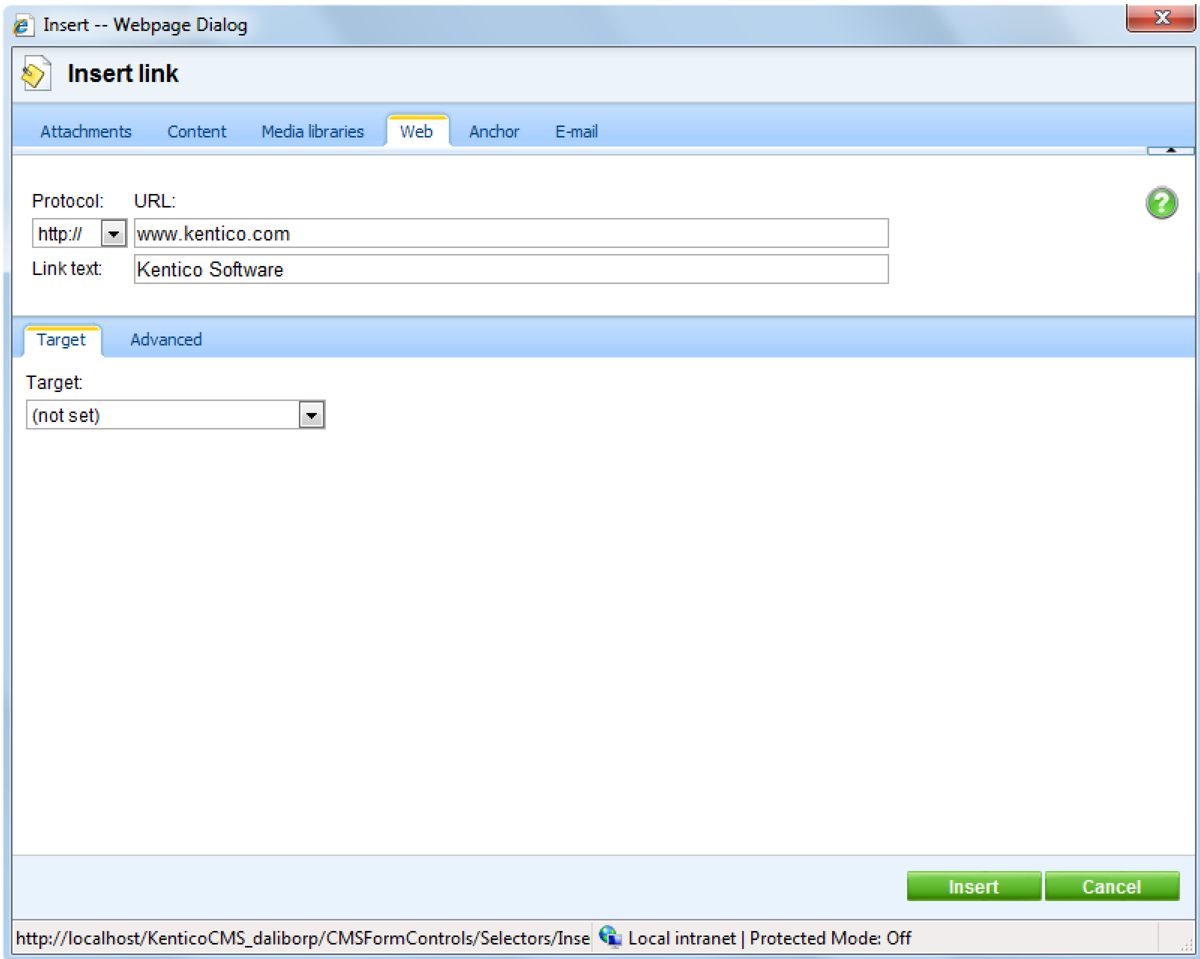


4.4.4.4 Links to Web

On the Web tab, you can link any resource on the Web by entering its URL. The output code is generated as the **<a>** HTML element:

```
<a href="http://www.kentico.com">Kentico Software</a>
```

On this tab, you can specify the same properties as described here, with the difference that the content of the **General** tab is displayed above the two other tabs.



4.4.4.5 Links to anchors

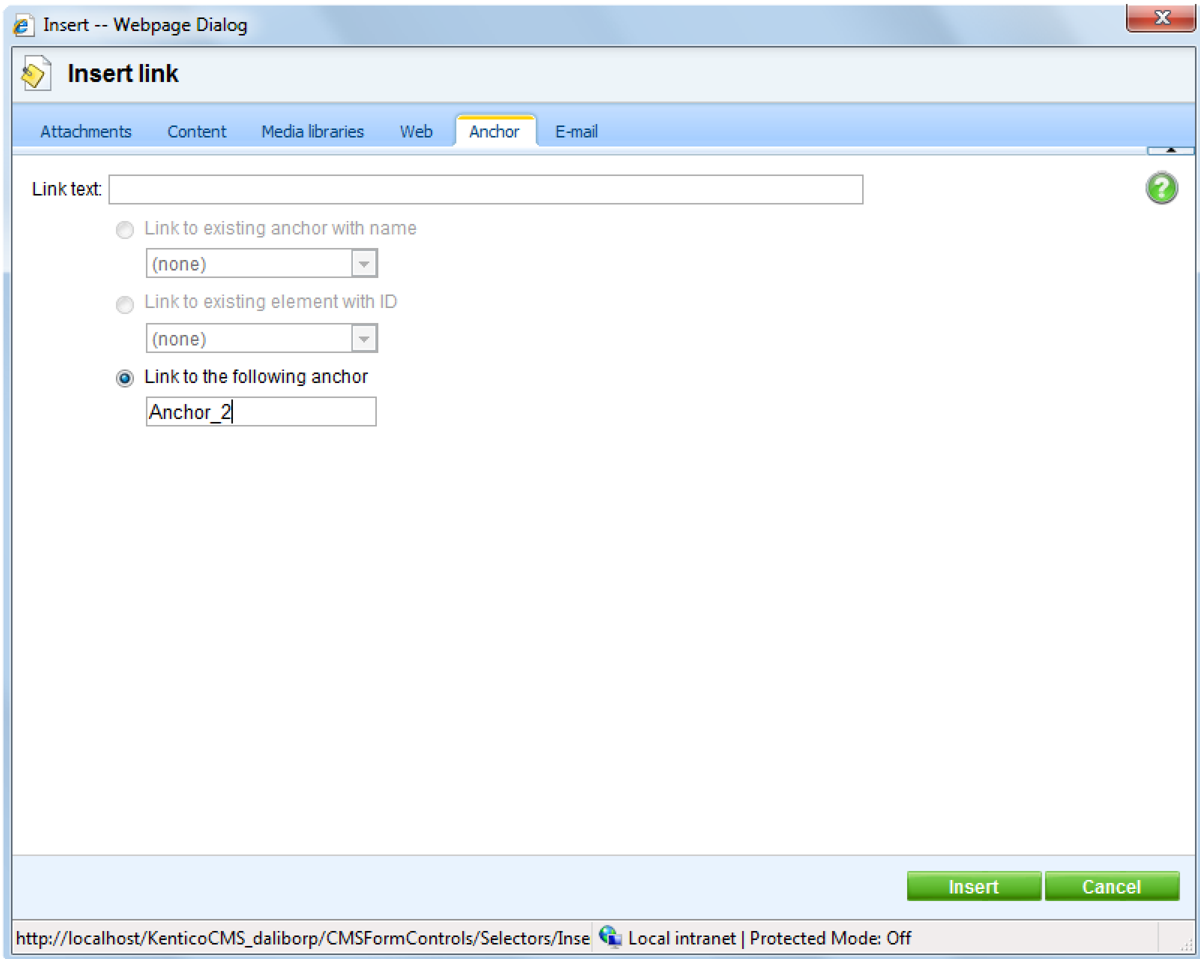
Via the Anchor tab, users can insert links to any anchor or any HTML element with specified ID attribute on the current page. Anchors are the <a> html elements with the **name** attribute specified. They can be inserted using the icon on the WYSIWYG editor toolbar. If you link to an anchor, the page will scroll to the anchor after clicking the link.

You have the following options on this tab:

- **Link text**: text of the link that will appear in the text; this field is visible only when inserting link into an empty space in the text area, i.e. when no text or object is selected
- **Link to existing anchor with name**: if selected, you can choose an anchor from the drop-down list below as the target
- **Link to existing element with ID**: if selected, you can choose a HTML element from the drop-down list below as the target
- **Link to the following anchor**: if selected, you can type in the name of the target anchor or ID of the target element manually

The output code looks like the following code sample, while the text after # is the name of the anchor or the value of the ID attribute:

```
<a href="#Anchor_2">Second chapter</a>
```



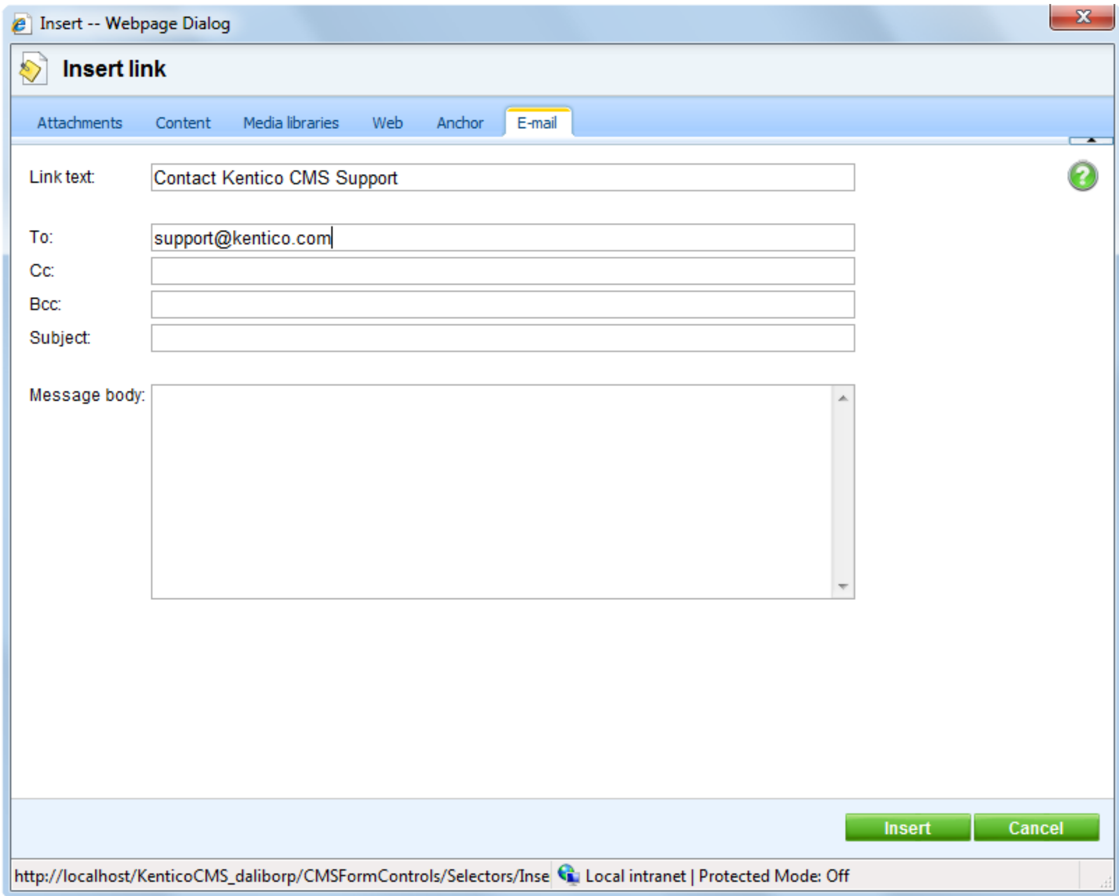
4.4.4.6 Mailto links

Via the E-mail tab, you can insert standard "mailto" links. After clicking such a link, the new message window of the user's e-mail program (e.g. Outlook) is opened, while some details may be pre-filled based on what is specified in the following properties:

- **Link text**: text of the link that will appear in the text; this field is visible only when inserting link into an empty space in the text area, i.e. when no text or object is selected
- **To**: e-mail recipient's address; required field; multiple addresses can be entered divided by semicolons
- **Cc**: e-mail copy recipient's address; multiple addresses can be entered divided by semicolons
- **Bcc**: e-mail blind carbon copy recipient's address; multiple addresses can be entered divided by semicolons
- **Subject**: subject of the e-mail
- **Message body**: text of the e-mail

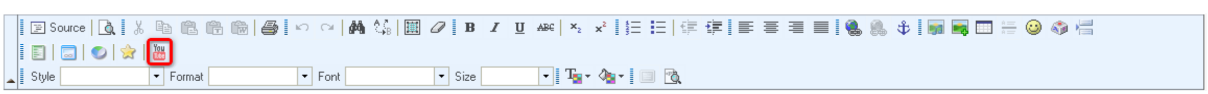
The output code looks as the following code sample:

```
<a href="mailto:support@kentico.com">Contact Kentico CMS Support</a>
```



4.4.5 Insert YouTube video

Using this dialog, video from YouTube can be easily added to a page. The dialog can be opened using the **Insert YouTube video** ( ) button on the WYSIWYG editor toolbar, as you can see in the screenshot below:



Inserting YouTube video

The general process of inserting a YouTube video is the following:

1. Place the cursor at the appropriate position in the text area and click the **Insert YouTube video** ( ) icon.
2. The dialog opens.
3. Insert the URL of the YouTube video into the **URL** field and click the icon.
4. The entered URL gets checked and if it is valid, default properties will be loaded and preview displayed.

5. Specify the video's properties according to your needs. The changes you make are reflected in the preview in the right part of the dialog.
6. When you are finished with the properties, click the **Insert** button.
7. The video is inserted into the text area.

YouTube video properties

The following properties can be specified in the dialog:

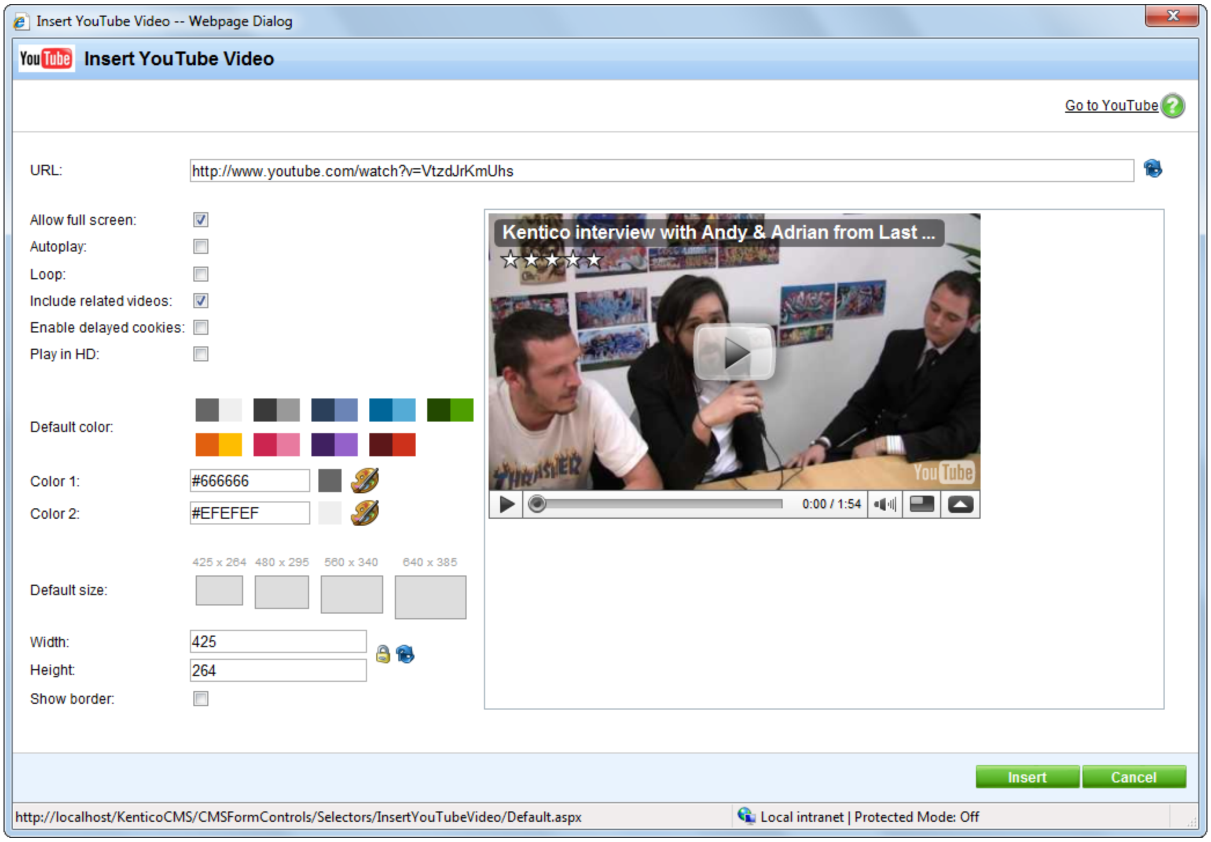
- **URL**: URL of the YouTube video; you can copy&paste it from your browser's address line or from the URL field on the video's page

- **Allow full screen**: indicates if the control to switch playback to full screen will be available in the video toolbar
- **Autoplay**: indicates if playback starts automatically when the page is loaded
- **Loop**: indicates if playback is continuously repeated in a loop
- **Include related videos**: indicates if related videos will be displayed when playback finishes
- **Enable delayed cookies**: indicates if delayed cookies should be used, read here for more information
- **Play in HD**: allows to play the video in HD by default; the user can switch back to normal quality by pressing the HD button while playing the video

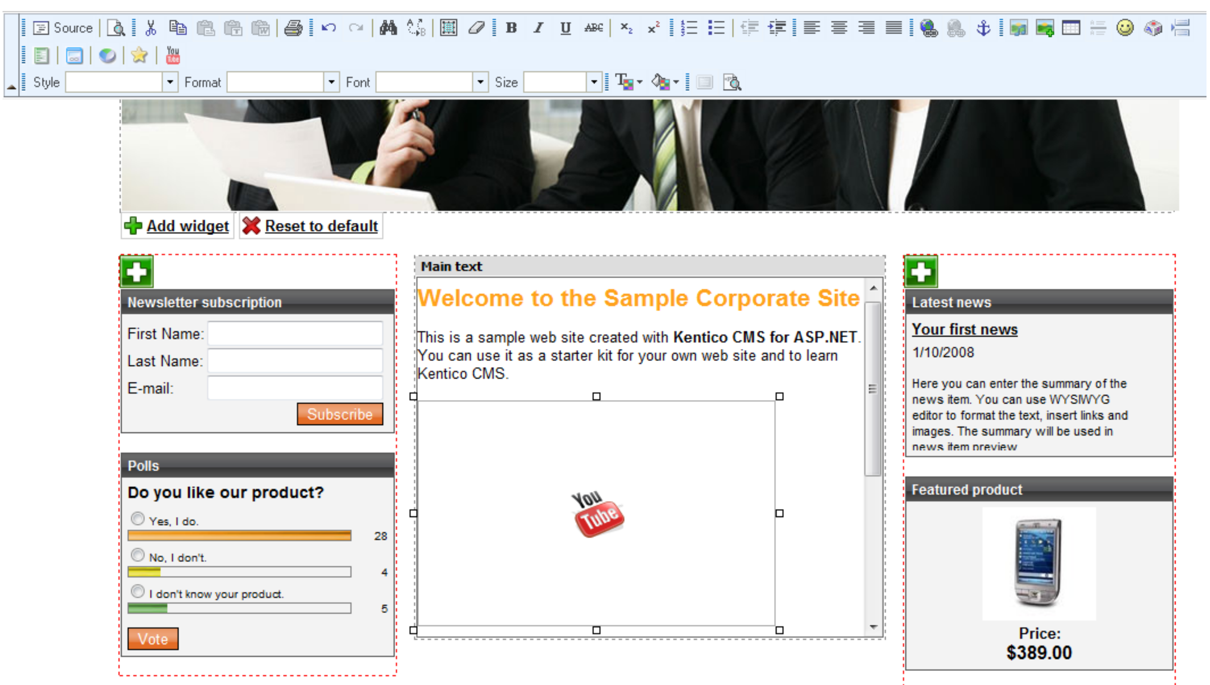
- **Default color**: you can choose one of the default color combinations which will be filled in the **Color 1** and **Color 2** properties
- **Color 1**: color of the border around the player, relevant only if the Show border property is enabled
- **Color 2**: color of the player toolbar

- **Default size**: you can choose one of the default sizes of the video player which will be filled in the **Widht** and **Height** properties
- **Width**: width of the video player
- **Height**: height of the video player
- **Show border**: indicates if border should be shown around the player; enabling this option adds 20px to both width and height of the player

This is what the dialog window looks like when a video is loaded:



In the WYSIWYG editor, the video itself is not displayed. You can only see a box with the YouTube logo in the middle to give you information about the player size:



YouTube video is generated as the **YouTubeVideo inline control**, as you can see in the code sample below:

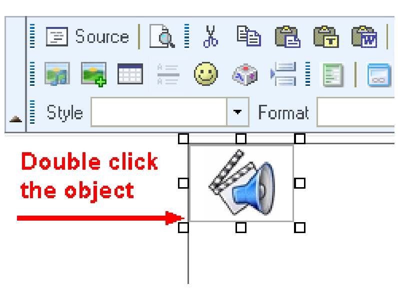
```
{^YouTubeVideo|(url)http://www.youtube.com/watch?v=VtzdJrKmUhs|(width)320|(height)
198|(fs)true|(autoplay)false|(loop)false|(rel)true|(cookies)false|(border)false|
(color1)#3A3A3A|(color2)#999999^}
```

On the live site, the video is displayed in the player:

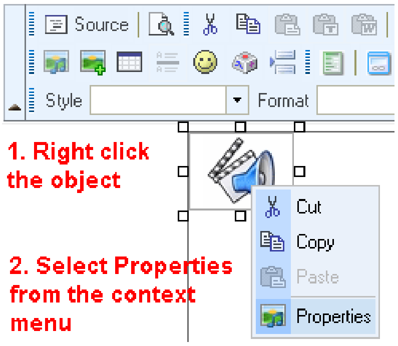
4.4.6 Editing inserted items

Properties of the images, audio, video, flash, links and YouTube video that are already inserted in the WYSIWYG editor can be edited. The same dialog that was displayed when you inserted the items can be opened again the following three ways:

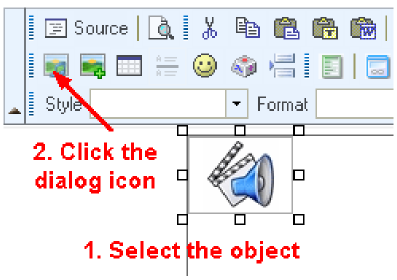
1. Double-click the item



2. Right-click the item and choose Properties from the context menu



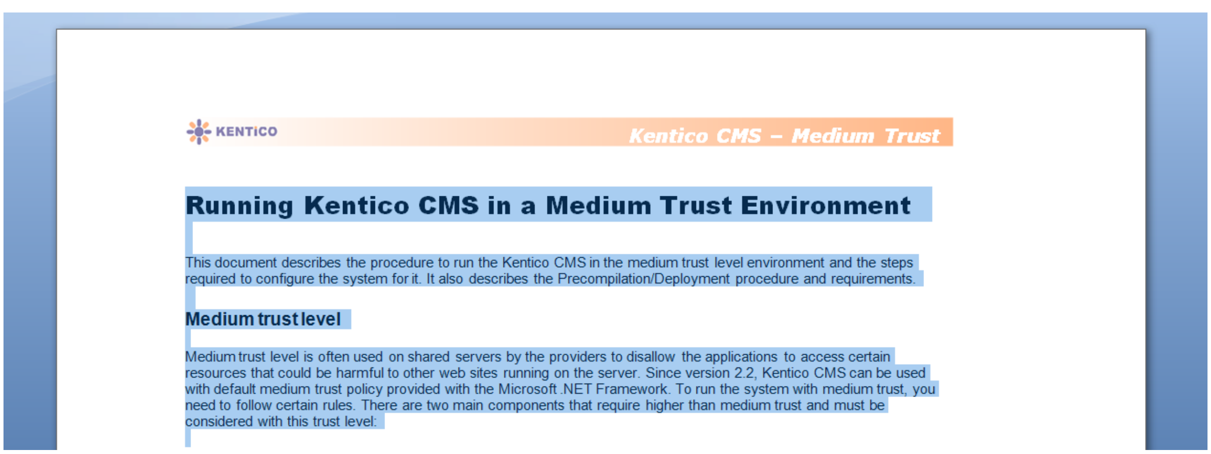
3. Select the item and click the appropriate button in the WYSIWYG editor toolbar



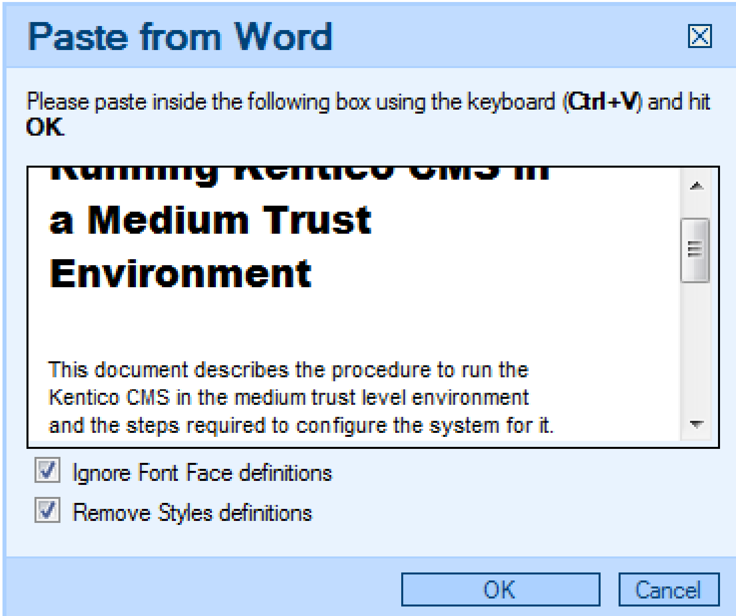
4.4.7 Copy & Paste from Microsoft Word

When, copying text from Microsoft Word, the text is encapsulated with lots of unnecessary tags that may break your web page design. Thus, the editor allows you to clean the pasted text so that it contains only basic formatting.

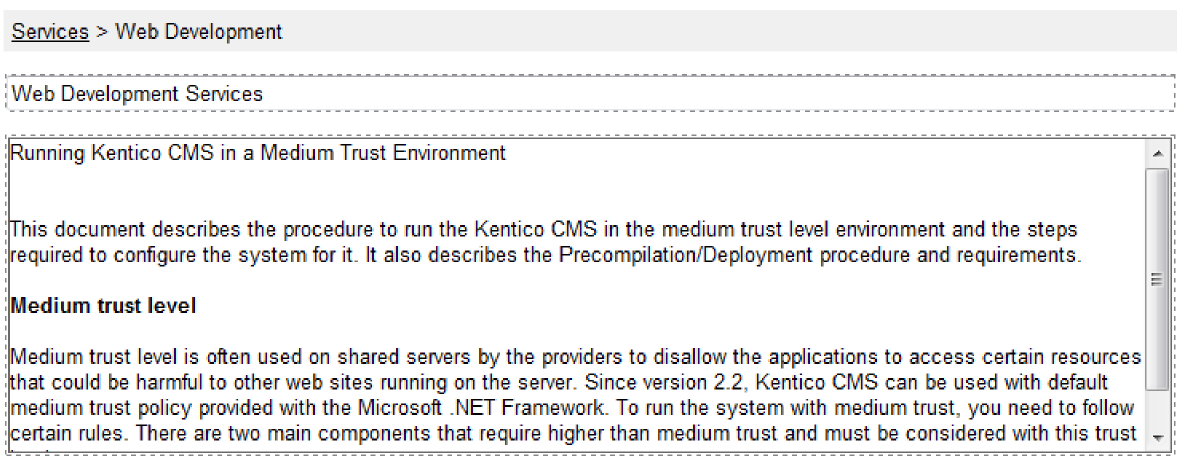
Select the text in Microsoft Word document and copy it to the clipboard (**CTRL+C**):



Place the cursor into the text in Kentico CMS editor and click Paste from Word. The **Paste from Word** dialog opens. Paste the text into the box using **CTRL+V** and check both boxes:



Click **OK** to paste the text. The pasted text looks like this:



As you can see, the style follows your website design. However, since Word doesn't provide appropriate tagging information, some formatting may not be preserved and you may need to apply the design manually - e.g. the header in the sample text.

Click **Save** to save the changes.

4.4.8 Defining custom toolbars

The toolbar of the WYSIWYG editor can be customized so that the content editors cannot use all the formatting features. It helps to keep the website design consistent and clean.

Defining the toolbar sets

You can define toolbar sets in **<web project>\CMSAdminControls\FCKeditor\fckconfig.js**.

You can define your own toolbar set by adding a command like this to the **fckconfig.js** file:

```
FCKConfig.ToolbarSets["Default"] = [
       ['Source', '-', 'Preview'],
       ['Cut', 'Copy', 'Paste', 'PasteText', 'PasteWord', '-', 'Print'],
       ['Undo', 'Redo', '-', 'Find', 'Replace', '-', 'SelectAll', 'RemoveFormat'],
       ['Bold', 'Italic', 'Underline', 'StrikeThrough', '-', 'Subscript',
'Superscript'],
       ['OrderedList', 'UnorderedList', '-', 'Outdent', 'Indent'/*, 'Blockquote'*/
],
       ['JustifyLeft', 'JustifyCenter', 'JustifyRight', 'JustifyFull'],
       ['InsertLink', 'Unlink', 'Anchor'],
       ['InsertImageOrMedia', 'QuicklyInsertImage', 'Table', 'Rule', 'Smiley',
'SpecialChar', 'PageBreak'],
       ['InsertBizForms', '-', 'InsertInlineControls', '-', 'InsertPolls', '-',
'InsertRating', '-', 'InsertYouTubeVideo'],
       '',
       '/',
       ['Style', 'FontFormat', 'FontName', 'FontSize'],
       ['TextColor', 'BGColor'],
       //['AutoSave'],
       ['FitWindow', 'ShowBlocks']       // No comma for the last row.
];
```

As you can see, the toolbar set definition consists of several arrays, such as **['Source','-','Preview']**. This array displays a group of two icons: Source and Preview, separated with a vertical line (defined by the '-' string).

You can insert a line break between the icon groups using the **'/'** string.

The code example above is the default toolbar set and contains all possible icons. You can create your custom toolbar sets by modifying it or using the icon names from it.

Save the changes you made to the **fckconfig.js**. Then, you need to clear the browser cache so that the file is updated in your browser.

Changes to the fckconfig.js file require clearing browser cache!

Every time you modify the fckconfig.js file (or any other file that is used for the WYSIWYG editor), you need to clear your browser cache so that the changes are applied. In Internet Explorer 7.0, you need to click **Tools -> Delete browsing history...** and click **Delete files...**

Then, you need to configure the page or document type so that it uses your new toolbar as described in the next paragraph.

Assigning the toolbar set to the page

If you want to assign the toolbar set to some **page with editable regions** (edited on the Page tab), you need to configure the web parts of type "Editable region" on the page template and set their Toolbar set property to the name of your toolbar set (in the example above, it's Default).

Toolbar set used for structured documents

If you want to modify the toolbar set used for **structured documents** (edited on the Form tab), you need to set the value of the Toolbar set property of the custom field. If you share the toolbar between multiple fields, the first field's toolbar set will be used.

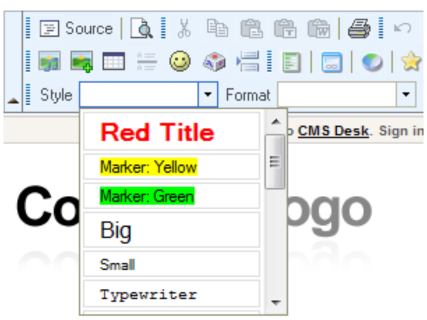
Standard toolbar sets

Kentico CMS uses some standard toolbar sets that are used by Kentico CMS modules:

- **Default** - used for structured documents
- **BizForm** - used for BizForm forms
- **Forum** - used for the WYSIWYG editor in forums (if enabled)

4.4.9 Defining styles

The WYSIWYG editor allows you to apply a chosen style to the text using the Style drop-down list in the editor toolbar:



The styles offered in the list are defined in file **<web project>\CMSAdminControls\FCKeditor\fckstyles.xml**. The structure of the file looks like this:

```
<Styles>
  <Style name="Image on Left" element="img">
    <Attribute name="style" value="padding: 5px; margin-right: 5px" />
    <Attribute name="border" value="2" />
    <Attribute name="align" value="left" />
  </Style>
  <Style name="Image on Right" element="img">
    <Attribute name="style" value="padding: 5px; margin-left: 5px" />
    <Attribute name="border" value="2" />
    <Attribute name="align" value="right" />
  </Style>
  <Style name="Bold text" element="span">
    <Attribute name="style" value="font-weight: bold;" />
```

```
  </Style>
  <Style name="Red text" element="span">
    <Attribute name="style" value="color: red;" />
  </Style>
  <Style name="Large text" element="span">
    <Attribute name="style" value="font-size: large;" />
  </Style>
</Styles>
```

Every style has some name and element it is used for. The styles are offered in the drop-down list based on the current position of the cursor. If you select some image, the styles for element img will be offered. If you select some text, the styles for element **span** or **div** will be offered.

If you choose to apply e.g. the **Red text** style to the following HTML code:

```
We provide web development services.
```

the result will be this:

```
<span style="color: red">We provide web development services.</span>
```

As you can see, the text was encapsulated with a **SPAN** element with attribute **STYLE** set to **color: red**.

You may want to apply CSS class names instead of hard-coded styles. In this case, your style definition will look like this:

```
<Style name="Green text" element="div">
  <Attribute name="class" value="GreenText" />
</Style>
```

and the result will be:

```
<div class="GreenText">We provide web development services.</div>
```

In your CSS stylesheet, you need to define the GreenText class name like this:

```
.GreenText { color: green; }
```

Changes to the fckstyles.xml file require clearing browser cache!

Every time you modify the fckstyles.xml file (or any other file that is used for the WYSIWYG editor), you need to clear your browser cache so that the changes apply. In Internet Explorer 7.0, you need to click **Tools -> Delete browsing history...** and click **Delete files...**

4.4.10 Dialogs configuration

When defining a new **HTML area (Formatted Text)** field, you can display advanced dialog settings by clicking the **Field advanced settings: Configure** link.

The following properties can be set, while they are organized in categories related to the particular tabs in the dialogs:

Content tab

- **Display tab**: indicates if the Content tab is displayed in the **Insert image or media** and **Insert link** dialogs
- **Available sites**: defines which sites' content trees will be available in the dialogs
- **Starting path**: alias path from where the content tree should be displayed; if not relevant for the selected site, its whole content tree will be displayed

Media tab

- **Display tab**: indicates if the Media libraries tab should be displayed in the **Insert image or media** and **Insert link** dialogs
- **Available sites**: defines which sites can the media libraries be selected from
- **Available site libraries**: defines which media libraries from the above selected site can be used
- **Available site groups**: defines which groups can be the media libraries selected from
- **Available group libraries**: defines which group libraries can be selected
- **Starting path**: path within the library from where the content should be offered; if not relevant for the selected library, the whole library will be offered

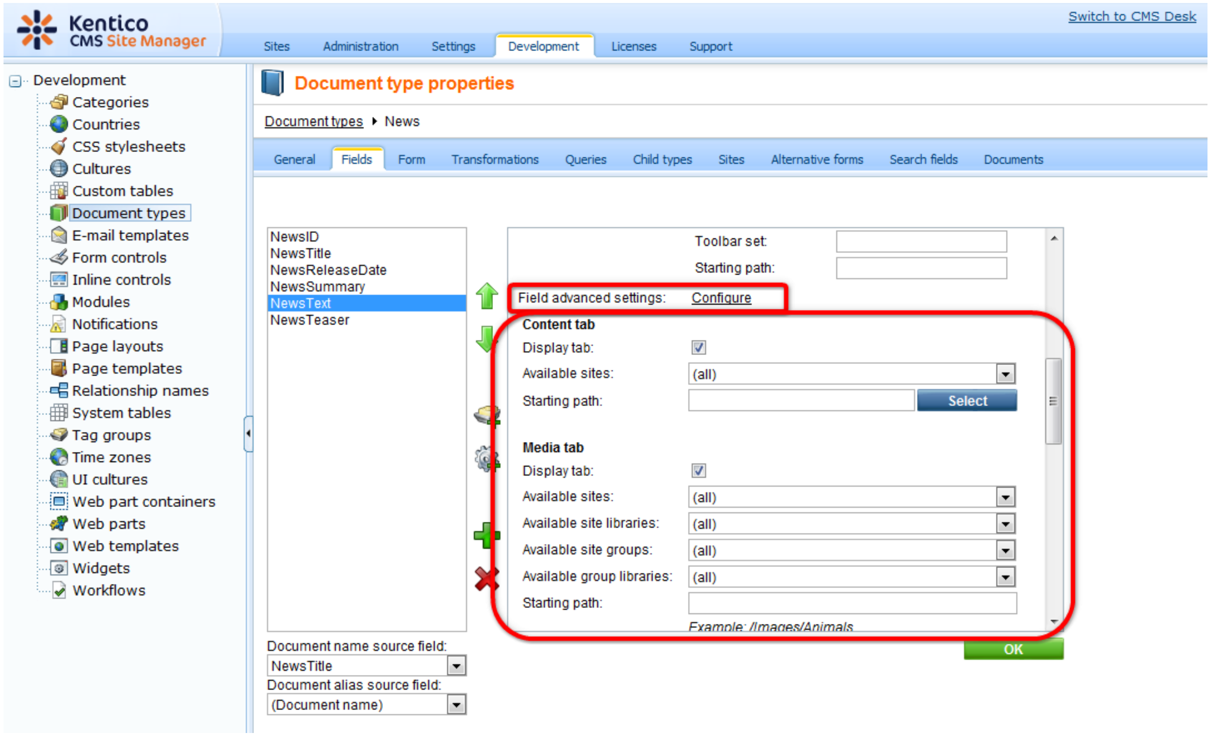
Other tabs

- **Display Attachments tab**: indicates if the Attachments tab should be displayed in the **Insert image or media** and **Insert link** dialogs
- **Display E-mail tab**: indicates if the E-mail tab should be displayed in the **Insert link** dialog
- **Display Anchor tab**: indicates if the Anchor tab should be displayed in the **Insert link** dialog
- **Display Web tab**: indicates if the Web tab should be displayed in the **Insert image or media** and **Insert link** dialogs

Automatic image resize on upload

- (**do not resize)**: if selected, images will not be resized and will be pasted in full size
- **(use site settings)**: if selected, images will be resized according to settings made in **Site Manager -> Settings -> Files**
- **(use custom settings)**: if selected, the settings specified in the three properties below will be used; the settings have the same effects as described here

Some of the properties have the **(current)** values (e.g. current site, current library, ...). This means that the current value from the page's context will be taken - e.g. if you are on a page belonging to some group and have the **Available group libraries** set to **(current group)**, the group to which the page belongs will be used.



Setup on last insertion

The dialog remembers the selection setup on last insertion for each user. This means that when a user opens the **Insert image or media** or **Insert link** dialog, all of the following are in the same state as on last insertion:

- selected tab
- selected view mode

Content tab only:
- selected site
- selected path in the content tree

Media libraries tab only
- selected site
- selected group
- selected media library
- selected path within the library

Dialogs-related settings

The following settings in the categories under **Site Manager -> Settings** are related to WYSIWYG editor dialogs:

- **URLs -> Use permanent URLs** - if enabled, URLs of documents and document attachments will be

generated in permanent format; if disabled, friendly URLs will be used
- **Media libraries -> Use permanent URLs** - indicates if links in permanent format should be used for files stored in media libraries
- **Files -> Automatic image resize on upload** - read here for more information on image resizing

4.4.11 Dialogs security

The following tables show which permissions are required to perform the listed actions on particular tabs.

Media libraries tab

In this table, you can see that three types of permissions may enable users to perform the required action:

Media library module permissions and Groups module permissions are those permissions that can be set in the permission matrices in Site Manager -> Administration -> Permissions.

Media library permissions are those permissions that can be set on the Security tab when editing a particular media library in CMS Desk -> Tools -> Media libraries.

Action/Permission	**Media library module permissions**		**Groups module permissions**			**Media library permissions**						
						File			**Folder**			**See content**
	Read	Manage	Read	Manage		Create	Delete	Modify	Create	Delete	Modify	
Global administrator in **group**/**global** media library												
Creating new media file	No permissions are required											
Creating new media folder	No permissions are required											
See media library content in dialogs	No permissions are required											
Group admin in **group** media library												
Creating new media file	No permissions are required											
Creating new media folder	No permissions are required											
See media library content in dialogs	No permissions are required											
Any other user in **global** media library												
Creating new media file		X			or	X						
Creating new media folder		X			or				X			
See media library content in dialogs	X											X
Any other user in **group** media library												
Creating new media file				X	or	X						

Creating new media folder				X	or				X			
See media library content in dialogs			X		or							X

Attachments tab - document already exists

To perform the following actions on the Attachments tab, the user needs to have the appropriate document permissions for the particular document.

Action/Permission	Document persmissions			
	Read	Create Manage	Modify	Delete
Creating new attachment (New file)			X	
Replacing source file of the attachment (Update)			X	
Deleting attachment (Delete)			X	
Moving attachment up or down (Move up/Move down)			X	
See attachments of the specified document	X			

Attachments tab - document doesn't exist yet

To perform the following actions on the Attachments tab, the user needs to have the appropriate document permissions for the particular document.

Action/Permission	Document persmissions			
	Read	Create Manage	Modify	Delete
Creating new attachment		X		
Replacing attachment		X		
Deleting attachment		X		
Moving attachment up or down		X		
See attachments of the specified document		X		

Content tab

To perform the following actions on the Content tab, the user needs to have the appropriate document permissions and the listed document types (CMS.File and CMS.Folder) need to be allowed to be created in the location.

The last two columns indicate if **cms.file** and **cms.folder** document types are **allowed child types** of the document under which it should be created. This can be set separately for each document type in Site Manager -> Development -> Document types -> edit the particular document type -> Child types tab.

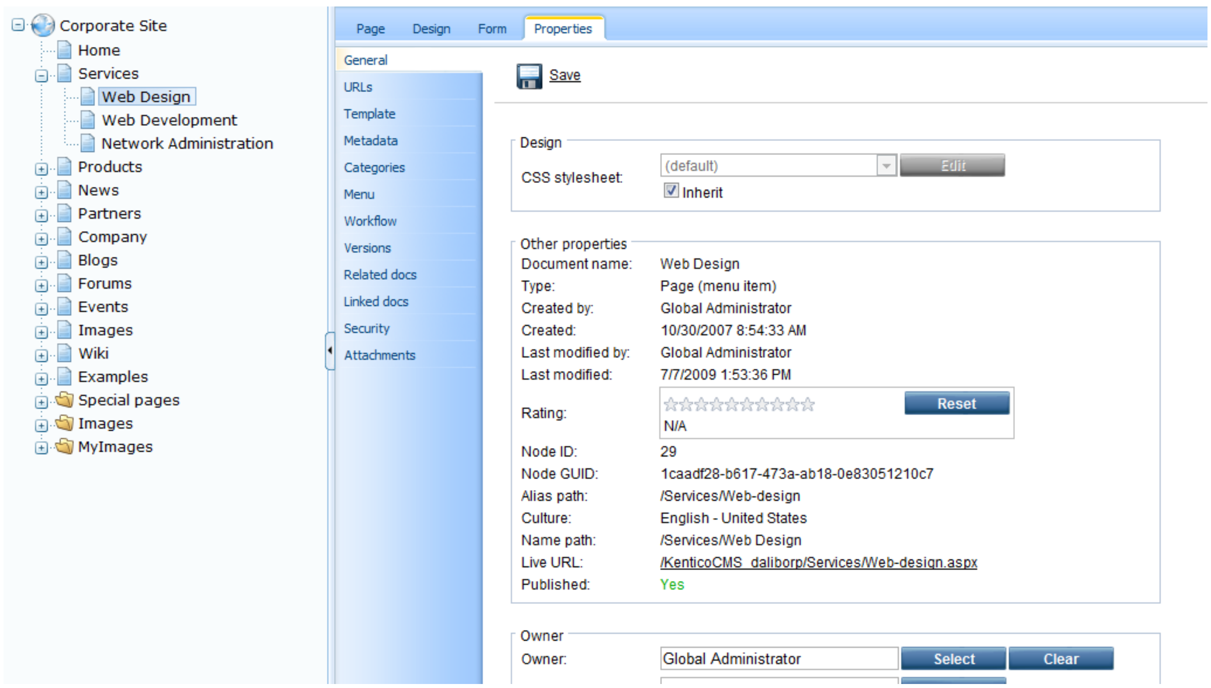
Action/Permission	Document persmissions					
	Browse tree	Create	Modify	Delete	CMS. File is allowed	CMS. Folder is allowed

		Manage				
Creating new cms.file		X			X	
Creating new folder		X				X
See child documents of the specified document	X					

4.5 Document properties

4.5.1 Overview

You can edit document properties in **CMS Desk -> Content**, after you click the document and click the **Properties** tab in the **Edit** mode:



4.5.2 General

General tab fields:

CSS Stylesheet	The CSS stylesheet used for the particular page. You can choose some particular stylesheet or use the default site stylesheet. You can also choose to inherit the stylesheet from the parent document.
Owner	Document owner is the user responsible for its editing. This feature doesn't imply any special permissions for the owner, but it allows for easier orientation in documents. The owner can see all their documents in **My Desk -> My Documents** section. The owner is by default set to the user who created the document. The owner can be changed only by users with "Modify permissions" permission.

Owned by group	This field shows which group is the owner of the document (this is particularly the case of group pages). This field is mainly informative, but in some special situations, you might find it useful to change the group that the document belongs to by using the Change button.
Rating	Rating of the document's content posted evaluated by site visitors; you can reset the value using the **Reset** button; see the Modules -> Content rating chapter for more details.
Cache	This option allows you to specify if the page content should be cached in memory (full-page caching). You can choose to inherit the settings from the parent document or to set a different value or to disable full-page caching.
Cache minutes	Determines how long should the content of this page be cached if caching is enabled.

4.5.3 URLs

URLs tab fields:

Document alias	The unique name of the document in the given section of the website. This name doesn't change when you modify the document name and it's used: 1. To define the **AliasPath**, which is the unique path to the document. The developers typically use the AliasPath in the Path property of the web parts and controls. 2. For the URL of the document. The URL of the document is like www.kentico.com**/products/cms**.aspx where /products/cms is the alias path of the document. **Forbidden characters** Some characters are forbidden in URLs and thus they are replaced with a safe character (by default, it's dash -). You can specify the forbidden characters and the replacement character in Site Manager -> Settings -> URLs, in values "Forbidden characters replacement" and "Forbidden URL characters".
Document URL path	If you want to use a specific URL for a given document that doesn't depend on the document AliasPath, you need to check the box **Use custom URL path** and enter the value, such as: **/cms-for-asp-net** The URL path must always start with /. Then, the document is accessible through two URLs: www.kentico.com**/products/cms**.aspx www.kentico.com**/cms-for-asp-net**.aspx The URL path is useful if you need to define a short URL of some page or if you need to optimize the URL for search engines. The URL path is, unlike the alias path, culture-specific. It means if you use URL paths: /product (for English version) /produkt (for German version)

	then when the user comes to /produkt.aspx, the language of the website is automatically switched to German.
URL extensions	Default URL extensions are defined in Site Manager -> Settings -> URLs -> Friendly URL extensions. If you check the **Use custom URL extensions** check box, you can define other extensions under which the document can be accessed. In case that the option is disabled, physical file extensions can also be used for **cms.file** documents. Multiple extensions can be entered divided by a semicolon. If you enter a semicolon without any extension, extensionless URLs will be allowed. Example: entering '.htm;;' for a document with URL path set to '/test' will cause that both <site path>/test.htm and <site path>/test will lead to the same page. For this to work, you also need to to set up your IIS for handling the 404 and 405 errors the same way as described here.
Track campaign	A web analytics campaign with the same name as entered here will be created on the first access to the page through the entered URL.
Document aliases	This section displays a list of other document aliases under which the document can be accessed. One document can have an unlimited number of aliases. You can edit ( ) and delete ( ) aliases in the list or create a new alias using the '**Add new alias**' link.

4.5.4 Template

In the Template dialog, you can specify the page template that will be used for the current document.

General properties

The page can be based on some specified page template or inherit the page template from the parent document.

Template	Current template name. Different template can be chosen after clicking the **Select** button.
Save as new template...	Saves the current template appearance as a new re-usable page template.
Inherit template	Indicates if page template should be inherited from the parent page.
Clone as ad-hoc template	Creates a new ad-hoc template based on the current page template. In many cases, you may want to re-use the ad-hoc page template for other pages. In this case, you need to save the ad-hoc template as a new re-usable template.
Edit template properties	Displays a new window with the template properties and settings.

Inherit content

This option allows you (developer) specify if and how the content should be inherited from the parent page(s). It overrides the content inheritance settings of the page template.

Inherit all	Document inherits all the content from all parent pages.
Do not inherit any content	Document does not inherit any content from the parent page.
Inherit only master page	Document inherits content from the first master page above the page in the content tree, i.e. if there are more master pages, it inherits only from the one which is the closest above.
Select inherited levels	Using this option, you can specify from which particular levels should the content be inherited.

4.5.5 Metadata

Here your can manage the information describing the page. You can choose to inherit the values from the parent document or enter document-specific values.

Page title	Title of the browser window (<title> element inner text).
Page description	Page description used for meta tags of the page and for searching content of the website.
Page keywords	Keywords used for meta tags of the page and for searching content of the website.
Page tag group	Tag group that will be used for tagging this document using the Page tags parameter below.
Page tags	Enter the tags that you want to tag the document with. When entering more than one tag into the **Page tags** field, the tags should be separated with a comma or a blank space. A combination of both in a single entry is also valid. The following examples are all valid entries for adding three tags - *tag1*, *tag2* and *tag3*: *tag1, tag2, tag3* *tag1 tag2 tag3* *tag1, tag2 tag3* In case that you are entering a tag consisting of more than one word, you should enclose it within quotation marks. Multiple long tags can also be entered and can be also divided by both blank spaces and commas: *"long tag1", tag, "long tag2"* *"long tag1" tag "long tag2"* Quotation marks can also be used for tags containing special characters that couldn't be used otherwise: *"tag@1", "tag#2", "long, strange: tag@#$"*

	The page tags field has also an insinuation function implemented. This functions offers you tags from the selected tag group while you are writing.

Global Settings

You can configure prefix of the page title, description and keywords for all documents on the website using the Site Manager -> Settings -> website dialog. Here you can configure the prefixes and also the standard format of the page title.

The default page title format is: {%prefix%} - {%pagetitle_orelse_name%}

It means that the format consists of the prefix followed by the page title value. If the page title value is not set, the document name is used.

Macro expressions in metadata

You can use macros in format {%ColumnName%} to insert the values of the current document into the title or other metadata fields. See Appendix A - Macro expressions for a list of possible macro expressions.

4.5.6 Categories

Categories are topic-related groups of documents. On this page, you can assign the selected document to an unlimited number of categories. There are two lists of categories:

- **My categories -** this list shows the current user's custom categories. Each of the categories can be edited ( ) or deleted ( ). By clicking the **New category** link, you can create your own category. The following details will be required:

Display name	Display name of the category.
Description	Text describing the category.

- **Global categories** - this list shows global categories that can be defined in **Site Manager -> Development -> Categories**. These categories can only be edited ( ) from this page.

In case that you want to assign the selected document to some of the categories, check the category's **Select** check-box in the list and click **Save**.

See the Modules -> Categories chapter for more details.

4.5.7 Menu

This dialog allows you to specify how the current document is displayed in the navigation.

General properties

Menu caption	The name of the document as it's displayed in navigation. It may be different to

	the document name. If no value is entered, the document name is used.
Show in navigation	Indicates if the document should be displayed in the navigation (in the menus). **Please note:** the document is displayed in the navigation if all of the following conditions are met: 1. The Show in navigation box is checked. 2. The document is published. 3. The type of the document matches the document types configured in the appropriate navigation control (web part) - by default, only **Page (menu item)** documents are displayed in navigation. 4. If you turn on the **Check permissions** property of the menu control, the user must be allowed to read the given document so that it appears in the navigation controls.
Show in site map	Indicates if the document should be displayed by the Site map web part (in the dynamic site map). **Please note:** the document is displayed in the navigation if all of the following conditions are met: 1. The Show in site map box is checked. 2. The document is published. 3. The type of the document matches the document types configured in the Site map control (web part) - by default, only **Page (menu item)** documents are displayed in navigation. 4. If you turn on the **Check permissions** property of the menu control, the user must be allowed to read the given document so that it appears in the navigation controls.

Menu action

You can choose from the following menu item behavior options:

Standard behavior	The menu item redirects the user to the page as expected.
Inactive menu item.	The menu item click doesn't raise any action - the item is disabled. This option is useful if you need to create a menu item for the main section in the drop-down menu that cannot be clicked, but the sub-items can be clicked.
Javascript command	If you enter some JavaScript command, it will be run when this menu item is clicked. Example: *alert('hello')*;*return false*;
URL redirection	The user is redirected to the target location when the menu item of this document is clicked. Example: *http://www.domain.com* or *~/products.aspx*

Menu item design properties

The menu item design properties are available in three alternatives:

- standard design

- mouse-over design - when you mouse-over the menu item
- highlighted design - style of the selected document

These values override the settings of the menu control (web part) and the CSS styles defined in the CSS stylesheet.

Please note: some of the following properties may not be applied to the menu control depending on the menu control you are using.

Menu item style	Style definition of the menu item. Values can be entered the same way as when defining a CSS class in a stylesheet. Sample value: *color: orange; font-size: 140%*
Menu item CSS class	CSS class defined in the website's stylesheet. Sample value: *h1*
Menu item left image	Image that will be displayed next to the menu caption on the left side. Sample values as below.
Menu item image	Image that will be displayed in the menu instead of the menu caption. You can enter either an absolute URL or a relative path in the content tree. Sample values: *http://www.domain.com/image.gif* *~/Images-(1)/icon.aspx*
Menu item right image	Image that will be displayed next to the menu caption on the right side. Sample values as above.

4.5.8 Workflow

On this tab, you can approve or reject document if it uses **workflow** and you're authorized to approve/reject given document in the given workflow step. See the Workflow and versioning chapter for more details.

Approve section

- **Comment** - you can add a comment which will be added to the e-mail message to the editor whose work you are approving
- **Send e-mail** - if the box is checked, a message can be sent to the editor whose work is currently being approved; if not, no message is sent to the editor

Workflow steps

This table displays the workflow steps of the current document's workflow, while pointing out the current step

Workflow history

This list displays the current document's workflow steps history.

4.5.9 Versions

On this tab, you can view the previous versions of the document if it uses a workflow. You can also roll back to a previous version of the document by clicking the icon, delete ( ) the older versions or destroy the whole document history.

When you click on the icon, you can preview the particular document version.

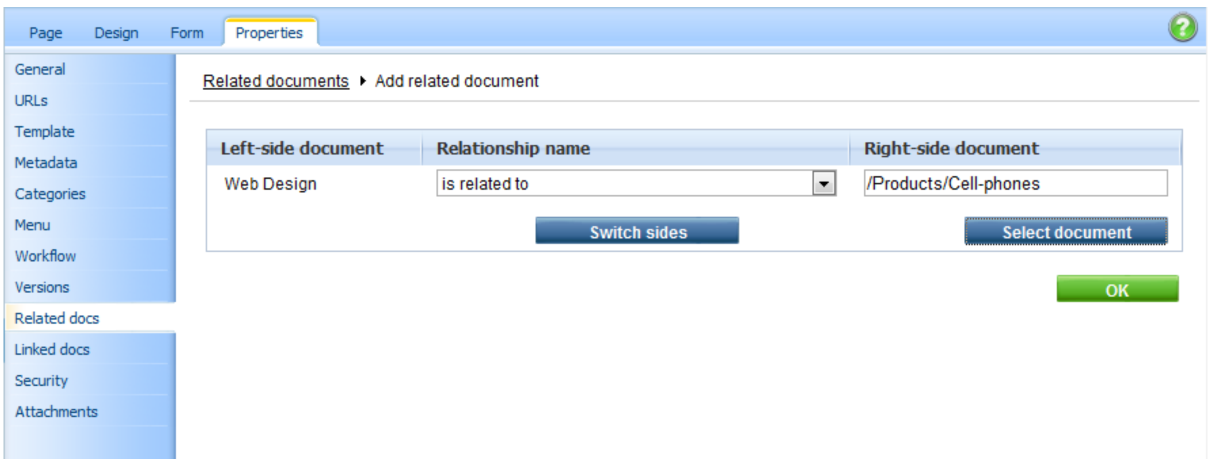
See chapter Workflow and versioning for more details.

4.5.10 Related docs

This dialog allows you to specify documents that are related to the selected document. Click ( ) **Add related document**.

Now you can:

- select the related document using the Select button
- switch the sides of the relationship
- select the type of the relationship



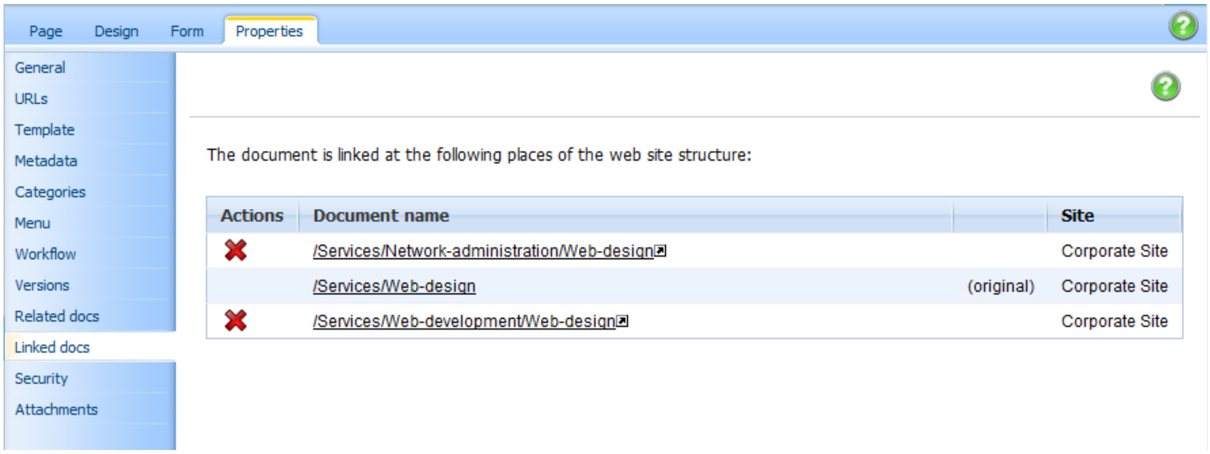
The related documents need to be displayed by a control (web part) that is configured for displaying related documents. If you're a developer, please see the **Kentico CMS Web parts and Controls reference -> Displaying related documents** for more details.

The relationship names need to be defined in **Site Manager -> Development -> Relationship names**. Only relationship names enabled for the current website are available in the drop-down list.

4.5.11 Linked docs

This dialog displays the linked documents (see Creating a linked document for explanation of the concept of linked documents).

You can delete the linked documents using the **Delete** ( ) icon or navigate to other linked documents or to the original document by clicking the link:



4.5.12 Security

This dialog is divided into two independent sections - **Permissions** and **Access**

Permissions

Here you can specify the document-level permissions for the given document. Please see the Authorization (permissions) chapter for more details.

Local and global permissions

Please remember that the local permissions are combined with global permissions defined for roles in the **CMS Desk -> Administration -> Permissions** dialog as described in the Authorization (permissions) chapter.

Access

Here you can configure if the page is accessible by public (anonymous) visitors or if it's only available to users who sign in with their user name and password. If you choose the option **Inherits**, the value is inherited from the parent document.

Please read the Secured website areas chapter for more details.

You can also specify if the give page or website section **requires SSL** (HTTPS) protocol. If you set it to Yes, the CMS automatically redirects the user to the URL with https:// protocol.

Requires authentication

- **Yes** - authentication is required to access the page
- **No** - authentication is not required to access the page
- **Inherits** - value of the setting is required from the parent page

Requires SSL

- **Yes** - users trying to access the page will be redirected to the HTTPS URL of the page
- **No** - users trying to access the page will access the page via the protocol from that they are comming (i.e. they will not be explicitly redirected to the secured URL, but if they are coming from a secured URL, they will access the page via secured URL)
- **Inherits** - settings of the parent page will be used
- **Never** - users trying to access the page will be explicitly redirected to the non-secured version of the page

4.5.13 Attachments

On this tab, files can be attached to the document.

On this page, you can see a list of the currently selected document's unsorted attachments. Grouped attachments can be added to the document on the Form tab in case that the appropriate field(s) are defined for the particular document type.

You can upload new attachment using the **New attachment** ( ) link. You can also perform the following actions with the attachments in the list:

- Using the **Delete** ( ) icon, you can remove the attachment from the document.
- Using the **Move up** ( ) and **Move down** ( ) icons, you can re-order the attachments. The order is stored in the **AttachmentOrder** property of each attachment. You can enter AttachmentOrder into the ORDER BY expression property of a displaying web part to have the attachments ordered accordingly.

Please note

The order of attachments is **not versioned** with documents' workflow. This means that if you change the order of attachments in one version of a document, the order is changed in all other versions too.

- Images have also the **Edit** ( ) icon available. This icon opens the image in the built-in image editor.
- Using the **Update** ( ) icon, you can replace the original attachment with a new one.
- After clicking an attachment's name, the attachment will be opened.

For more information on the Attachments module, please refer to the Document attachments chapter of this guide.

4.5.14 Languages

This tab is displayed only only for multilingual sites with Translation management enabled. See the Multilingual and international support -> Translation management chapter for more details.

On this tab, you can view the language versions of the selected document. The language versions can have the following statuses:

The colors indicate the following translation statuses:

- **Translated** - the document is translated and up-to-date

- **Outdated** - the document is translated but outdated, which means that the default language version has been modified (or published when using workflow) more recently than the translated version
- **Not available** - the document's version in the language does not exist

By clicking the **Edit culture version** () icon, you can be redirected to the **Edit -> Page** mode of the selected culture version. By clicking the **Add new culture version** () icon of a not-available version, you can proceed to creating the new culture version of the document.

4.6 File management

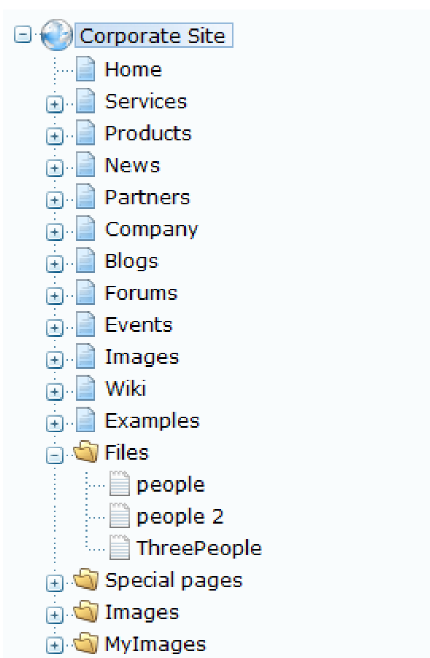
4.6.1 File management overview

Kentico CMS allows you to upload files (such as GIF, JPG, SWF, PDF, XLS, DOC, etc.) to the Kentico CMS database or file system and manage them as any other content.

File management approaches

There are four types of media files from the file management perspective:

- **CMS.File documents** - these files are uploaded by the content editors as new *CMS.File* documents into the content tree. You will typically use this type for files that are used as part of unstructured documents, such as document links or images inserted into editable regions of pages. It is advisable to have files stored within folders (*CMS.Folder* document type). You can also use the File import module when uploading multiple files. The Document library module allows convenient management of CMS.File documents on the live site.



- **Document attachments** - these files are stored as a part of a structured document and their life cycle is also bound with the document (including workflow and versioning). You can have an unlimited number of files attached to a document. A detailed description of the whole concept and examples of typical usage can be found in the Document attachments sub-chapter.

- **Media Libraries** - the Media Libraries module allows storage of large amounts of files, while large file sizes are supported; the whole module and its typical usage is described in Modules -> Media libraries.

- **Unmanaged files** - these files are part of the website theme and they should be stored in the *<web project>\app_themes\<theme>\images* folder on the disk. They usually include images and Flash animations used throughout the site. These files are not managed by the CMS system.

For performance optimization tips related to file management, please refer to the Development -> Caching and performance -> File management and performance chapter of this guide.

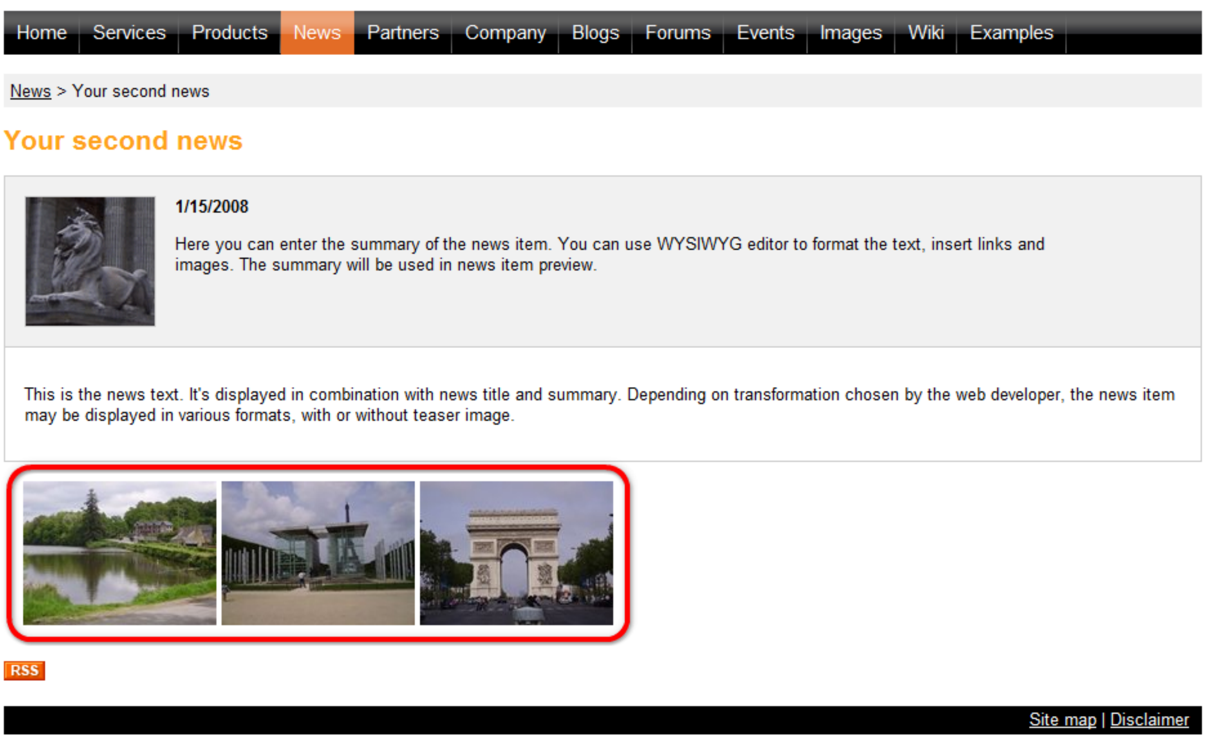
4.6.2 Document attachments

4.6.2.1 Overview

Document Attachments are a concept of attaching files to documents in Kentico CMS. This concept greatly simplifies the way you work with files. Each document can have any number of attached files, which is a great step forward compared to the previous use of the File field, which allowed only one uploaded file per one defined File field.

Attachments are directly bound to a document's life cycle. So if a document gets published, its attachments get published too. If you delete a document, its attachments will also be deleted.

You can find a live example of document attachments on the sample **Corporate Site**. If you enter the **News** section and display some of the news items, you can see the images at the bottom of the page, as highlighted in the screenshot below. These are the attachments of the news document displayed by the **Attachment image gallery** web part.



There are two types of attachments:

- **Unsorted** - added via the Properties -> Attachments tab; using this approach, you can easily add attachments to any document; all attachments added this way are taken as **one group** and displayed together using the web parts
- **Grouped** - added by defining a 'Document attachments' field for a document type and then uploading a file on a document's **Form** tab; for each Document attachments field, you can have an unlimited number of files attached; you can also define more fields of this type to have several groups of attachments which can be displayed separately on the live site
- **File field** - another type of field that can be used for file upload via the **Form** tab; the difference from Grouped attachments is that only one file can be uploaded into one File field

The attachments can be displayed with the document using one of the following web parts:

- **Attachment image gallery** - using the default transformation, the web part displays thumbnails of document's image attachments; after clicking a thumbnail, lightbox will open, displaying the attachment in a full size; for non-image attachments, file type icon will be displayed instead, while after clicking it, users are able to download the file; of course, you can modify this behaviour by defining your own transformation
- **Document attachments** - displays a list of document's attachments; after clicking an attachment, it is opened in a new window or offered for download in case that it is not an image
- **Attachments data source** - data source web part for providing attachments to a connected displaying web part; see also Using DataSource web parts

More information about the web parts can be found in this chapter.

You can also display document's unsorted attachments using the following two inline controls that can be added via the **Editable text** web part:

- **Attachment image gallery** - displays the document's unsorted attachments' thumbnails; after clicking the thumbnail, the attachment will be displayed in a lightbox
- **Document attachments** - displays the document's unsorted attachments; after clicking the thumbnail, image attachment will be displayed on a new page and non-image attachments will be offered for download

More information about the inline controls can be found in the Available inline controls chapter.

Settings which can be configured in **Site Manager -> Settings -> Files** are applied to document attachments. You can learn about available settings in the Files-related settings chapter.

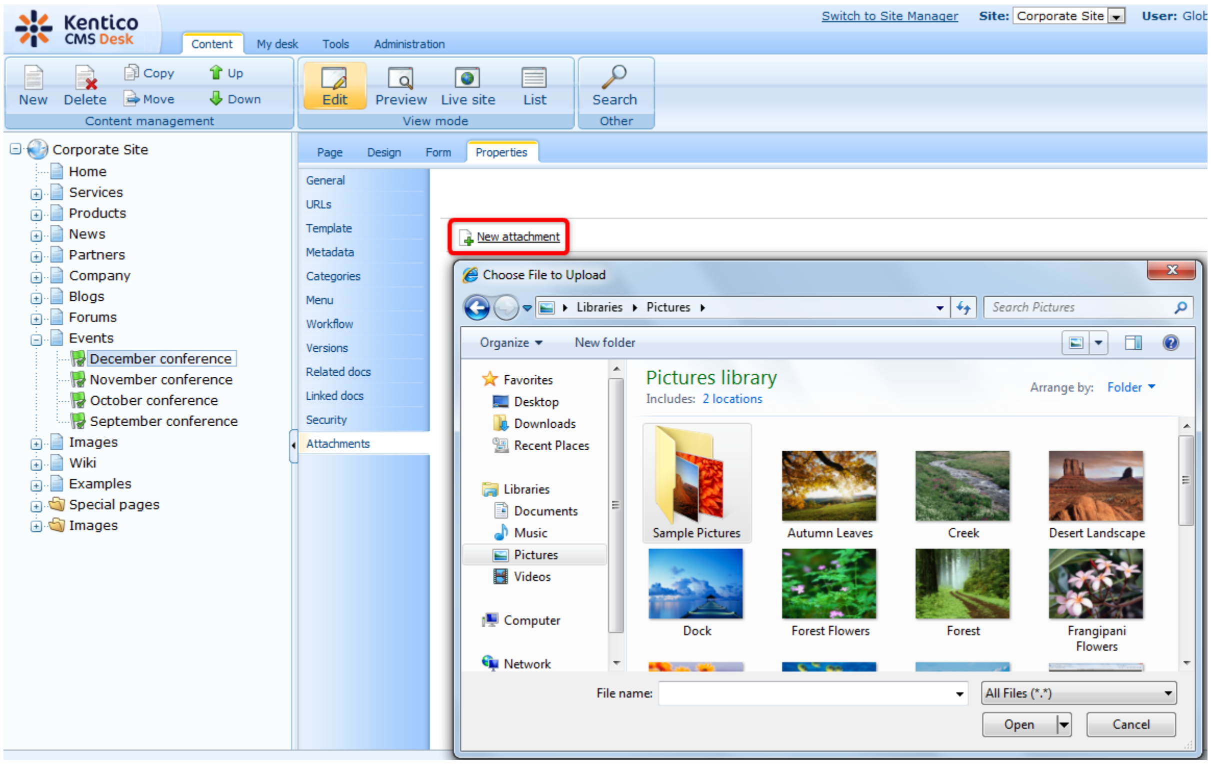
4.6.2.2 Example: Unsorted attachments

In the following example, you will learn how to add attachments to a document via the **Properties -> Attachments** tab and display them on the live site. We will use the **Events** section of the sample **Corporate Site**. First, we will upload the attachments to some of the events in the section. Then we will add the **Attachments image gallery** web part to the **Events** page, which will display the attachments for each displayed event.

1. Adding attachments to a document this way is quite simple. First, you need to select a document from the content tree and switch to its **Properties -> Attachments** tab.

2. On the tab, click the **New attachment** ( ) link. The familiar **Choose file** dialog will be displayed. Choose any file from your local drive and click the **Open** button.

Please note: The mouse pointer doesn't change when you hover the **New attachment** ( ) link. This is by design and you needn't be worried about it.



3. Repeat the same procedure so that you have a few files attached to at least one document. Preferably, include some images among them. The document's attachments tab should look similarly to the screenshot below after the attachments' upload. Note that if you hover an image in the list, its thumbnail is displayed as in the screenshot.

In the list of attachments, you can do several things with each attachment:

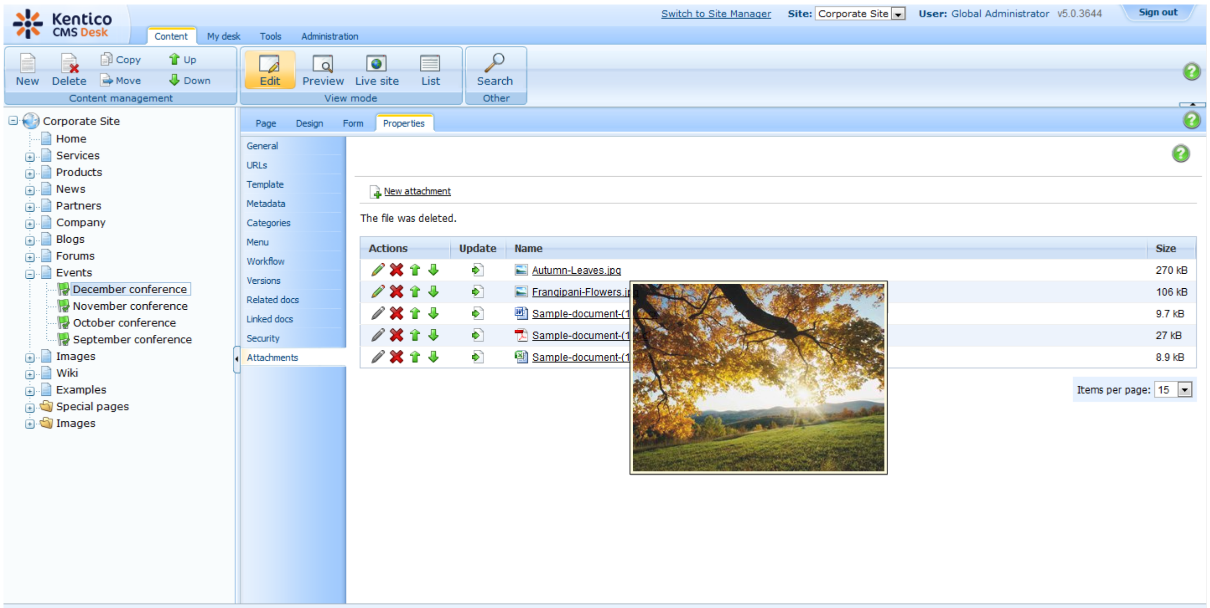
- Using the **Delete** ( ) icon, you can remove the attachment from the document.
- Using the **Move up** ( ) and **Move down** ( ) icons, you can re-order the attachments. The order is stored in the **AttachmentOrder** property of each attachment. You can enter AttachmentOrder into the ORDER BY expression property of a displaying web part to have the attachments ordered accordingly.

Please note

The order of attachments is **not versioned** with documents' workflow. This means that if you change the order of attachments in one version of a document, the order is changed in all other versions too.

- Images have also the **Edit** ( ) icon available. This icon opens the image in the built-in image editor.
- Using the **Update** ( ) icon, you can replace the original attachment with a new one.

- After clicking an attachment's name, the attachment will be opened.

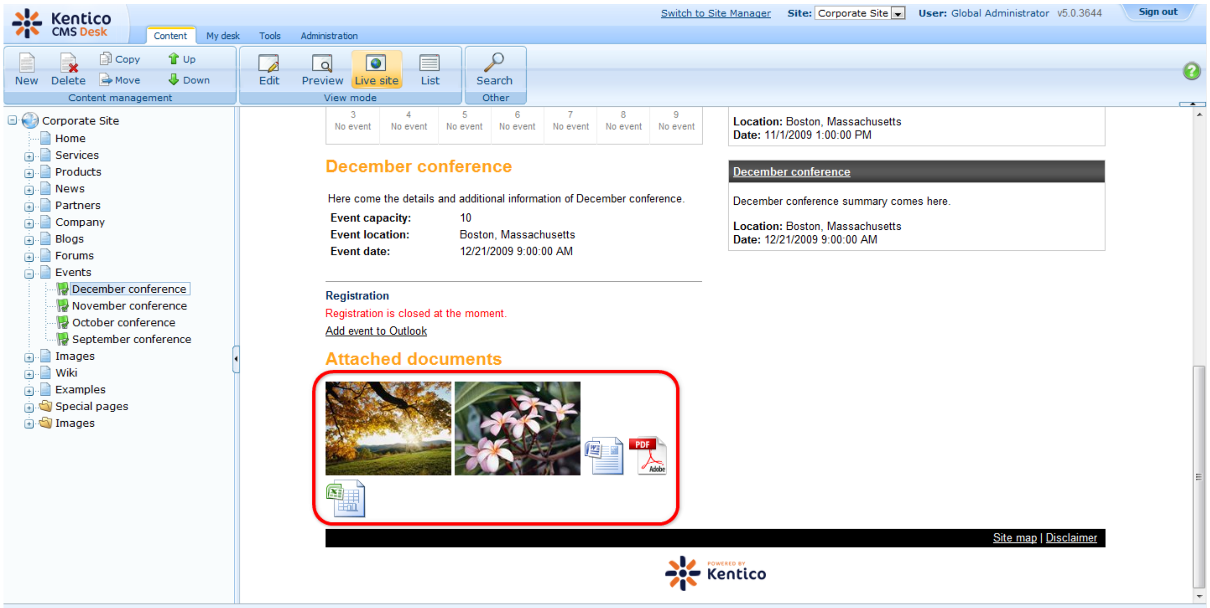


4. Now that we have the attachments uploaded, we can display them on the live site. The events are displayed by a repeater on the **Events** page, so we will have to add the **Document attachments** web part to this page.

Select the **Events** page from the content tree and switch to the **Design** tab. Click the **Add web part** ( ) icon of **zoneLeft** web part zone and choose the **Attachments -> Attachments image gallery** web part. You do not need to set any web part properties in order for the web part to display the attachments. However, you can add some heading via the Content before property:

- **Content before**: <h1>Attached documents</h1>

Click **OK**. If you switch to the live site now, you should see its attachments' thumbnails as in the screenshot below.

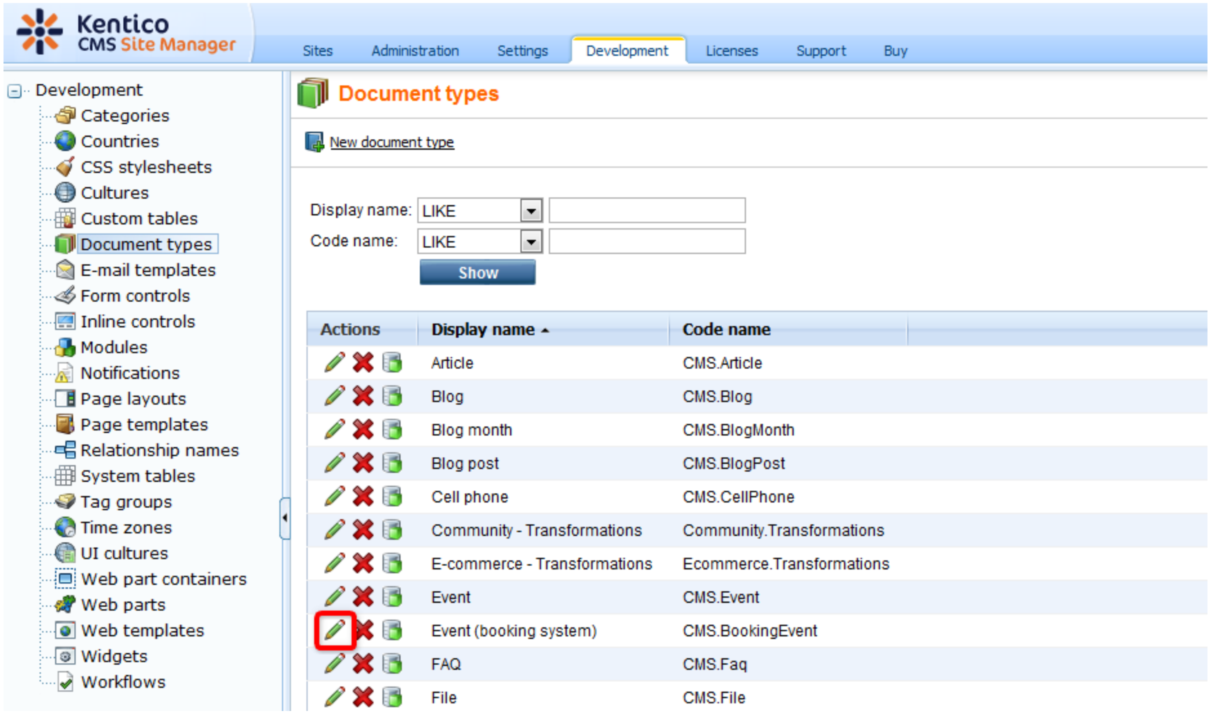


4.6.2.3 Example: Grouped attachments

In the previous example, you learned how to add attachments to a document via the **Properties -> Attachments** tab. This is the easier approach, however, in some situations, you may want to have more groups of attachments.

In this example, you will learn how to add two **Document attachments** fields to a document type and how to use them on the live site. Again, we will use the **Events** section of the sample **Corporate Site**. So if you completed the previous example right before, please delete the **Document image gallery** web part from **zoneLeft** in order to get the page to the original appearance.

1. The events' document type is **CMS.BookingEvent**. We will need to add the fields to it. Go to **Site Manager -> Development -> Document types** and choose to **Edit** (✏) the **CMS.BookingEvent** document type.

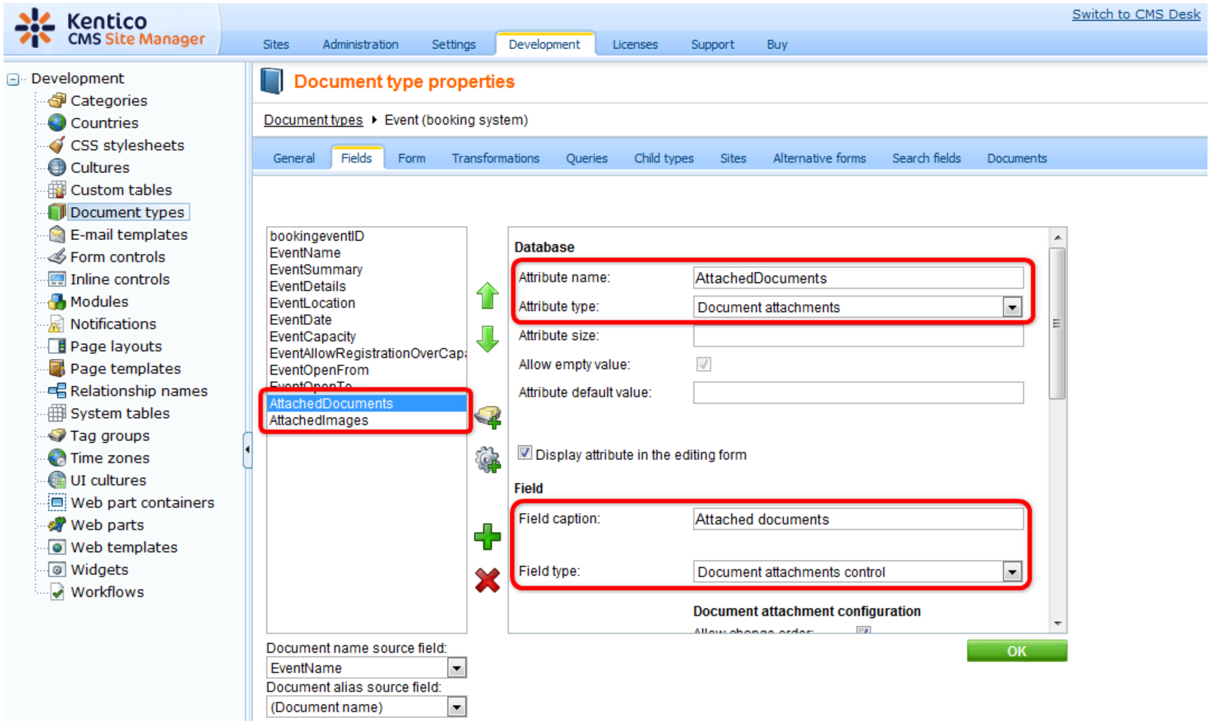


2. Switch to the **Fields** tab. Use the **New attribute** () icon to add the following two attributes. For each attribute, set only the following values and leave the default values for the rest.

- **Attribute name**: AttachedImages
- **Attribute type**: Document attachments
- **Field caption**: Attached images
- **Field type**: Document attachments control

- **Attribute name**: AttachedDocuments
- **Attribute type**: Document attachments
- **Field caption**: Attached documents
- **Field type**: Document attachments control

Finally, use the **Move up** () and **Move down** () arrows to move the two attributes to the bottom of the list so that they will be displayed at the bottom of the **Form** tab.



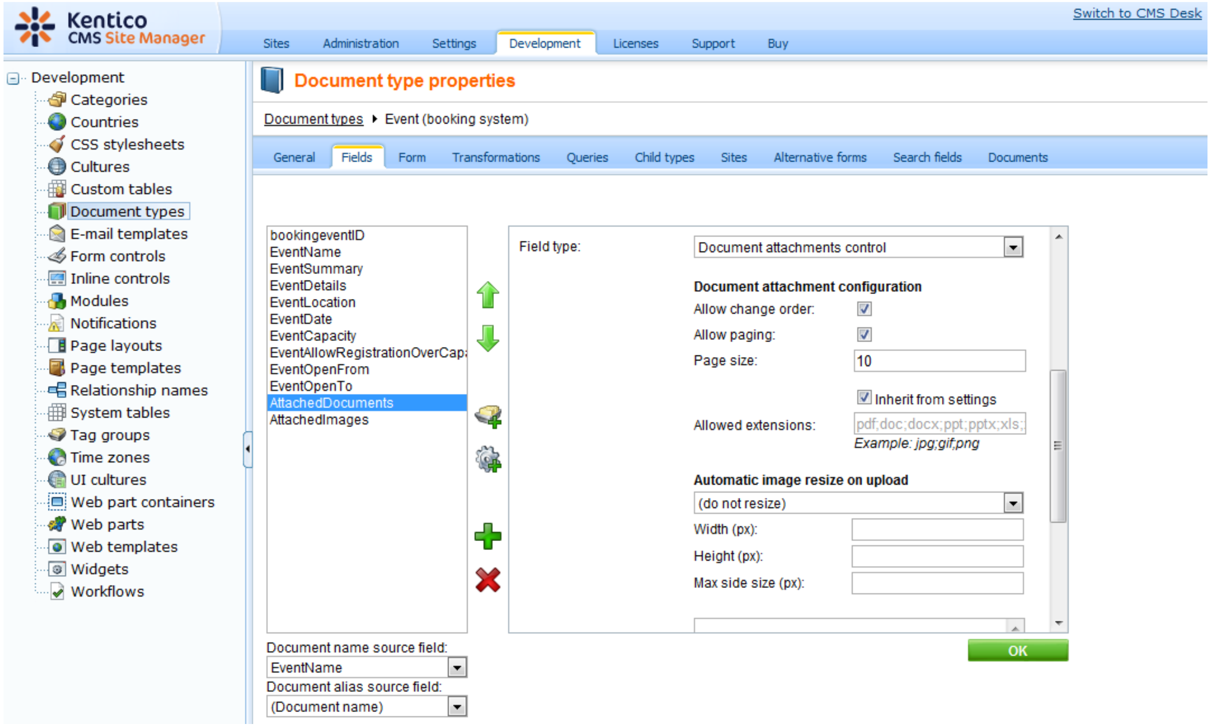
3. Optionally, you can set the following specific properties of the field:

- **Allow change order** - indicates if attachments can be re-ordered
- **Allow paging** - indicates if paging should be enabled for the list of attachments
- **Page size** - page size applied if paging is allowed
- **Allowed extensions** - allowed extensions of files to be uploaded; files with other extensions will not be uploaded; check Inherit from settings to use values specified in Site Manager -> Settings -> Files -> Upload extensions

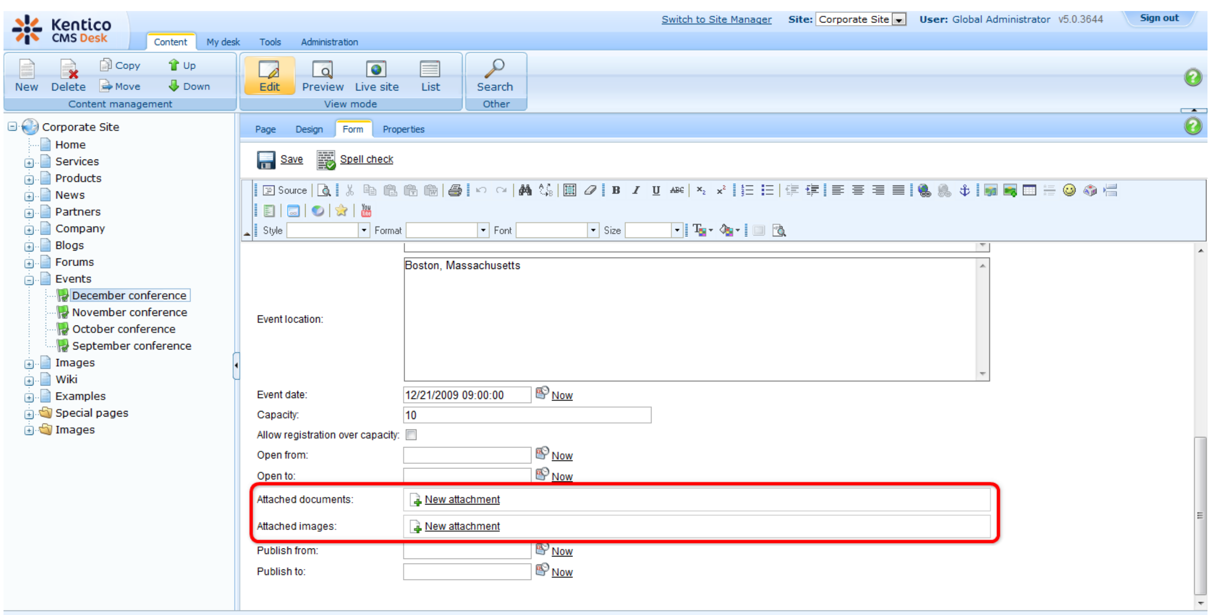
- **Automatic image resize on upload**:
  - **(do not resize)** - uploaded images will not be resized
  - **(use site settings)** - uploaded images will be resized according to site settings in Site Manager -> Settings -> Files -> Automatic image resize on upload
  - **(use custom settings)** - uploaded images will be resized according to the Width, Height and Max side size values set below

  Depending on which values you fill in the **Width**, **Height** and **Max side size** fields, the functionality is the following:

  - **Only width or only height** - images will be resized so that the width/height matches the entered value; the other dimension is also resized so that the aspect ratio is kept
  - **Both width and height** - images will be resized so that both dimensions match the entered values; the aspect ration needn't be kept in this case
  - **Max side size** - if one of the image's sides is larger than this value, the image will be resized so that its larger side's dimension matches the entered value; the aspect ratio is kept and width and height settings are not applied



4. Now that we have the two fields defined, we can upload some attachments to the existing events on the site. Switch to **CMS Desk** and select the **Events -> December conference** from the content tree. Switch to its **Properties -> Form** tab. If you scroll down the page, you should see the two fields defined in the previous step.



5. Use the **New attachment** ( ) link to add attachments into both fields. According to the name of the field, upload some documents into the first one and some images into the second one.

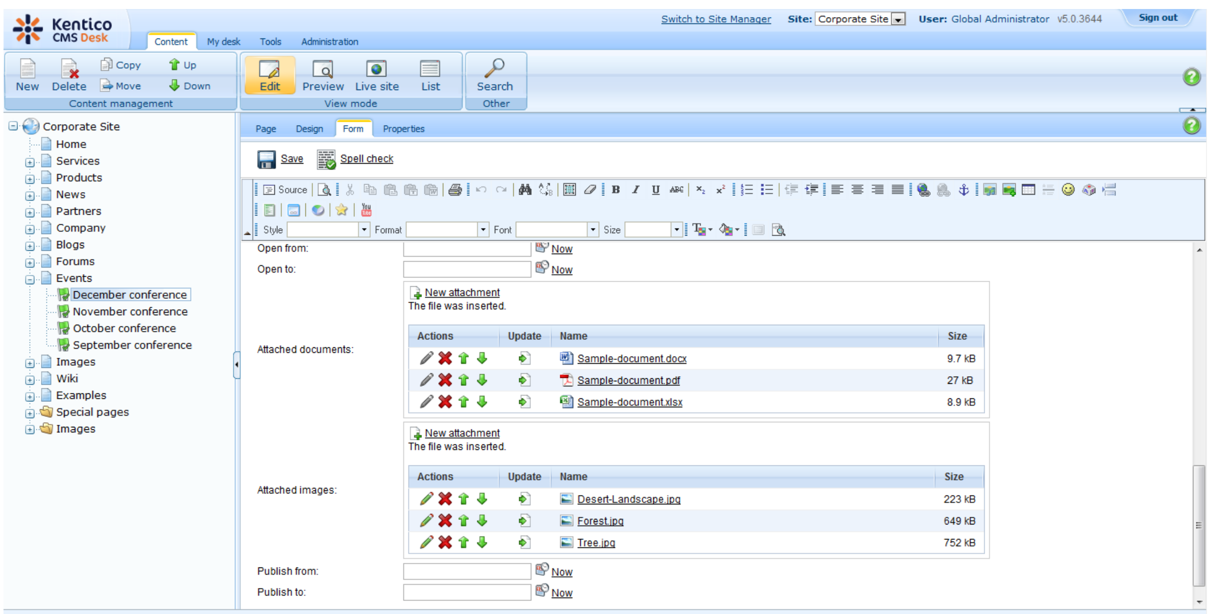
Just as on the **Properties -> Attachments** tab, you can perform the following actions with the attachments in the lists:

- Using the **Delete** (✖) icon, you can remove the attachment from the document.
- Using the **Move up** (⇧) and **Move down** (⇩) icons, you can re-order the attachments. The order is stored in the **AttachmentOrder** property of each attachment. You can enter AttachmentOrder into the ORDER BY expression property of a displaying web part to have the attachments ordered accordingly.

Please note

The order of attachments is **not versioned** with documents' workflow. This means that if you change the order of attachments in one version of a document, the order is changed in all other versions too.

- Images have also the **Edit** (✎) icon available. This icon opens the image in the built-in image editor.
- Using the **Update** () icon, you can replace the original attachment with a new one.
- After clicking an attachment's name, the attachment will be opened.



6. Now that we have the attachments uploaded, it's time to have them displayed on the live site. The events are displayed by a repeater on the Events page, so we will have to add the displaying web parts to this page. First, we will add the **Document attachments** web part. We will configure it so that it will display attachments from **Attached documents** field.

Select the **Events** page and switch to the **Design** tab. Click the **Add web part** (✚) icon of **zoneLeft** web part zone and choose the **Attachments -> Document attachments** web part. Set the following properties of the web part, leave the rest at the default values.

- **Show for document types**: CMS.BookingEvent
- **Attachment group**: Event (booking system) -> Attached documents

- **Content before**: <h2>Attached documents</h2>

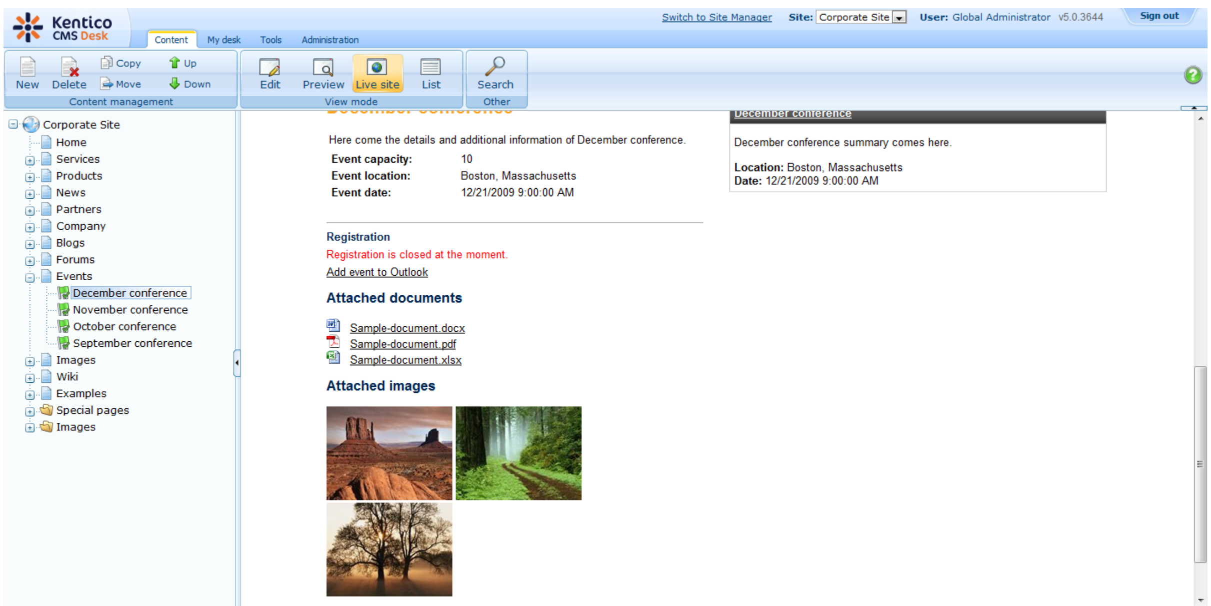
Click **OK**.

7. Below the documents, we will want the attached images to be displayed. For the images, it will be better to use the **Attachment image gallery** web part. Click the **Add web part** (➕) icon of **zoneLeft** web part zone and choose the **Attachments -> Attachment image gallery** web part. Set the following properties of the web part, leave the rest at the default values.

- **Show for document types**: CMS.BookingEvent
- **Attachment group**: Event (booking system) -> Attached images
- **Content before**: <h2>Attached images</h2>

Click **OK**.

8. If you switch to the live site now, you should see the two web parts displaying the attachments, each web part one group defined earlier in this example. If you add attachments to any other event, they would be displayed with the event too.



4.6.2.4 Available web parts

In this chapter, you can see an overview of the web parts that can be used to display document attachments. Only the most important web part properties are explained here. For a complete list of web part properties, please refer to [Kentico CMS Web Parts] reference guide or click the **Documentation** link at the top right corner of the web part properties window.

Document attachments

Sample-document-(1).docx
Sample-document-(1).pdf
Sample-document-(1).xlsx

The web part is located in the **Attachments** web part category. This web part displays a list of files attached to a document. After clicking an attachment, it is opened in a new window.

The following properties of the web part are the most important:

- **Path**: alias path of the document whose attachments should be displayed; if blank, the currently displayed document's attachments will be displayed
- **Attachment group**: specifies the field from which the attachments should be displayed; if blank, ungrouped attachments from the Properties -> Attachments tab will be used
- **Transformation**: transformation used for displaying the attachments; the default transformation is *CMS.Root.AttachmentList*

Attachments image gallery

The web part is located the **Attachments** web part category. The web part is particularly useful for image attachments. It displays thumbnails of images attached to the document. After clicking a thumbnail, the clicked image is displayed in its full size by a **lightbox**. For non-image attachments, its file type icon is displayed. After clicking such an icon, the file is offered for download.

The following properties are the most important:

- **Path**: alias path of the document whose attachments should be displayed; if blank, the currently displayed document's attachments will be displayed
- **Attachment group**: specifies the field from which the attachments should be displayed; if blank, ungrouped attachments from the Properties -> Attachments tab will be used
- **Transformation**: transformation used for displaying the attachments; the default transformation is *CMS.Root.AttachmentLightbox*
- **Selected item transformation**: transformation used to display the selected item; the default transformation is *CMS.Root.AttachmentLightboxDetail*
- **LightBox Configuration**: the properties in this section can be used to customize the used LightBox

Attachments Data Source

This is a special web part - a data source. It is not visible on the live site. It only provides data to another connected web part (e.g. Basic Repeater), which will display the provided data. More information on data source web parts can be found in this chapter.

4.6.2.5 Available inline controls

There are two inline controls that can be used for displaying documents' unsorted attachments. The inline controls can be added to a page via the **Editable text** web part, using the WYSIWYG editor on the **Page** tab. To add the control to the page, just place the cursor in the appropriate position and click the **Insert Inline Control** ( ) icon. You can choose one of the following two controls for displaying attachments:

Attachment image gallery

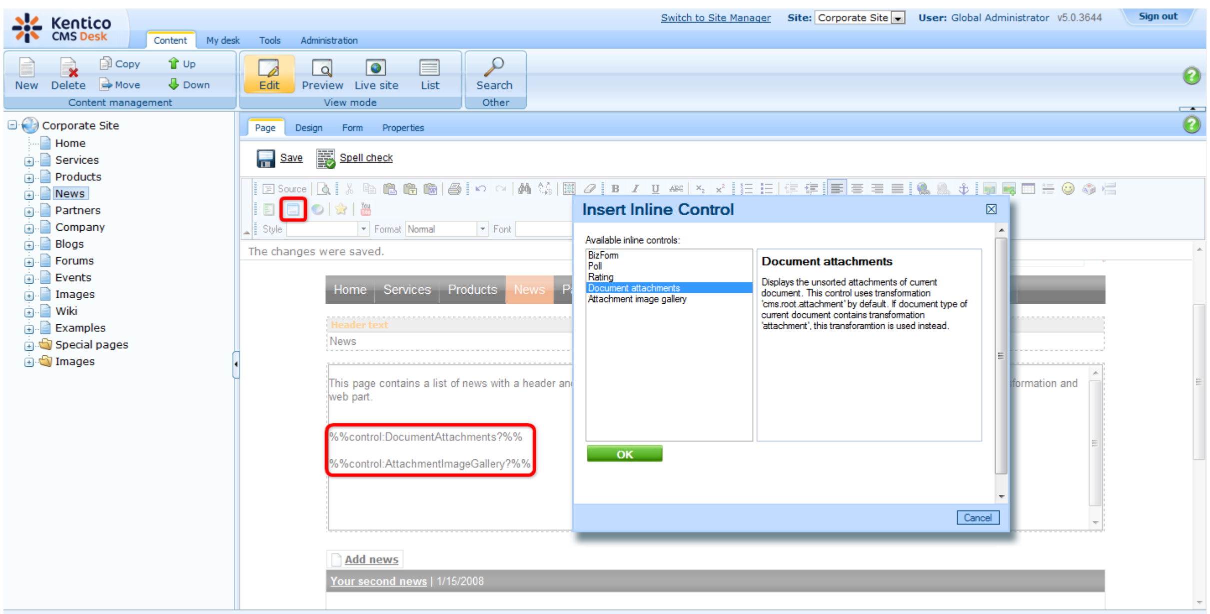
Displays the document's unsorted attachments' thumbnails based on the **cms.root. attachmentLightbox** transformation. After clicking the thumbnail, the attachment is displayed in a lightbox based on the **cms.root.attachmentLightboxDetail** transformation.

In the Editable text's text area, the control is reperesented by the following string: *%%control: AttachmentImageGallery?%%*

Document attachments

Displays the document's unsorted attachments' thumbnails based on the **cms.root.attachment** transformation. After clicking the thumbnail, image attachment is displayed on a new page and non-image attachments are offered for download.

In the Editable text's text area, the control is reperesented by the following string: *%%control: DocumentAttachments?%%*



4.6.2.6 Handling attachments in transformations

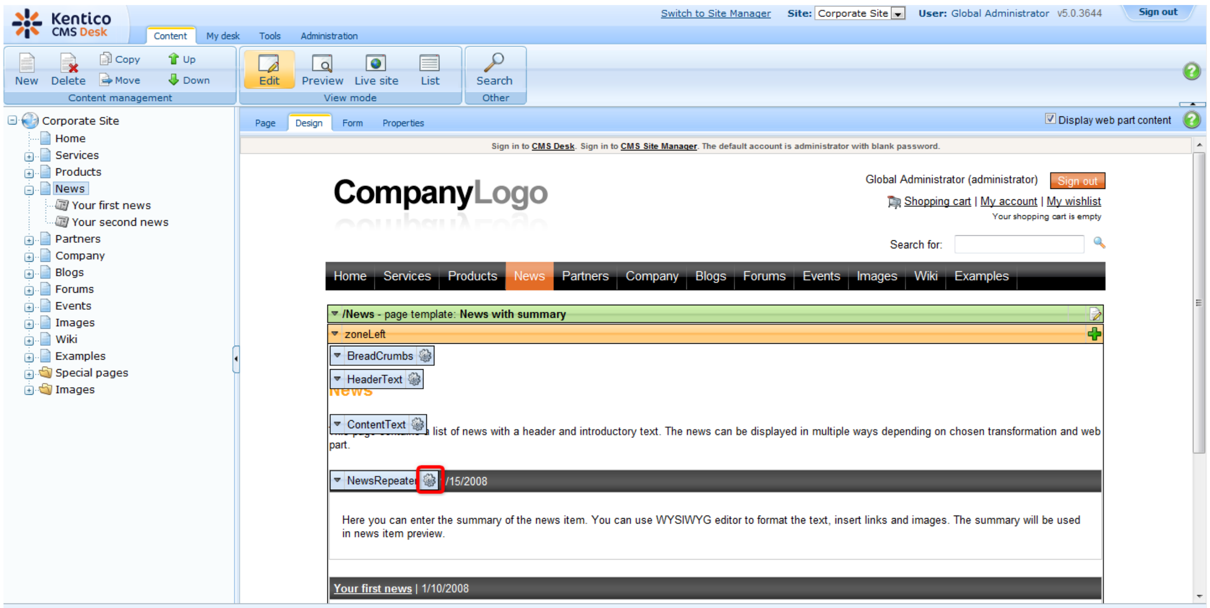
You can display document's unsorted attachments by adding one of the following two controls directly into the transformations:

- **~/CMSAdminControls/Attachments/DocumentAttachments.ascx**
- **~/CMSAdminControls/Attachments/AttachmentLightboxGallery.ascx**

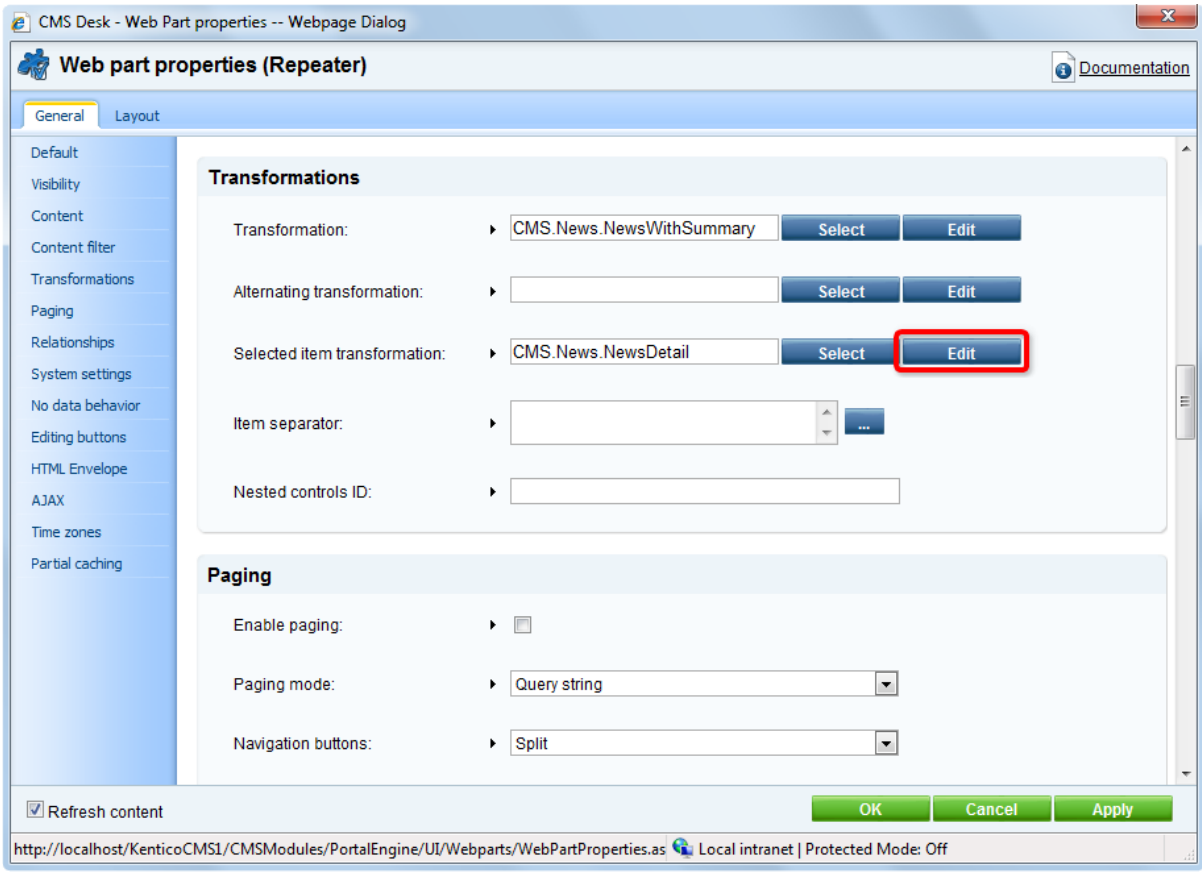
In the following example, you will learn how to use these controls in transformations. The news in the **News** section of the sample **Corporate Site** have some images attached by default. In this example, you have learned how to display these attachments using a web part. Another way of displaying the attachments is by modifying the particular transformation for the document type.

The news are displayed on the News page using a **Repeater**. Detail view of each news item is displayed using the **CMS.News.NewsDetail** transformation. We will modify this transformation so that it displays the attachments along with the news summary and text.

1. Go to **CMS Desk** and select the **News** page from the content tree. Switch to **Design** tab and choose to **Configure** ( ) the **NewsRepeater** web part.



2. In the web part properties window, choose to **Edit** the **Selected item transformation**.



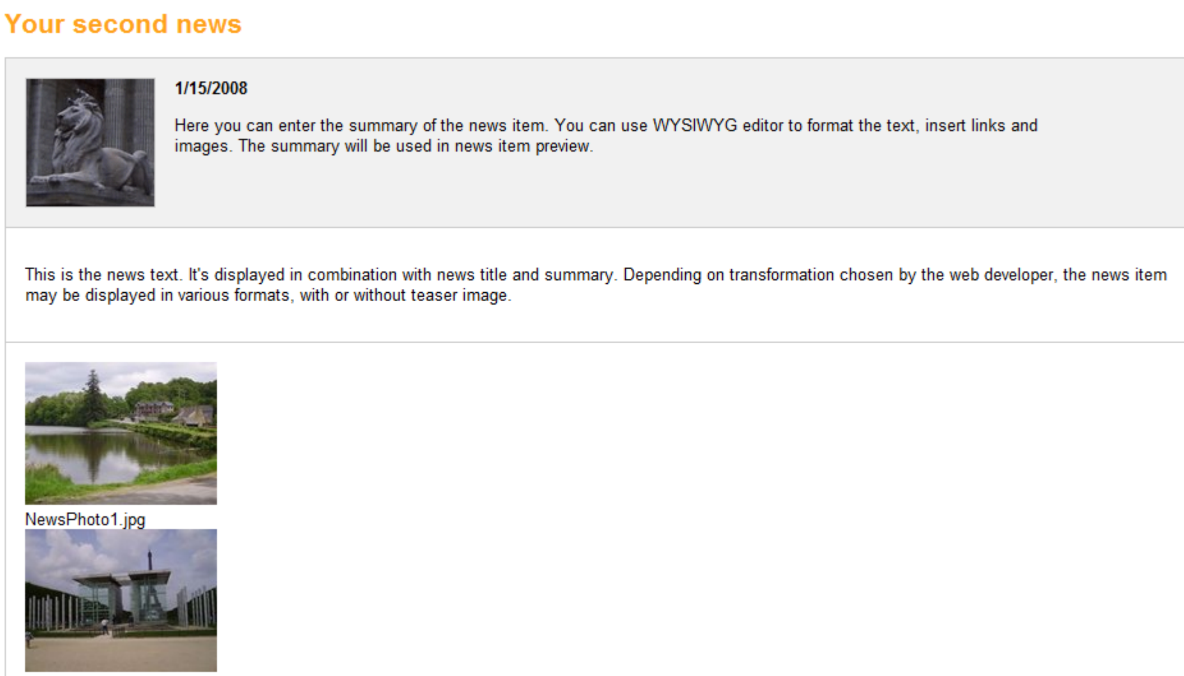
3. Replace the original transformation with the following code, which is the original transformation with the highlighted parts added.

```
<%@ Register Src="~/CMSInlineControls/DocumentAttachments.ascx" TagName
="DocumentAttachments" TagPrefix="cms" %>

<div class="newsItemDetail">
<h1><%# Eval("NewsTitle") %></h1>
<div class="NewsSummary">
  <%# IfEmpty(Eval("NewsTeaser"), "", GetImage("NewsTeaser")) %>
  <div class="NewsContent">
    <div class="Date"><%# GetDateTime("NewsReleaseDate", "d") %></div>
    <%# Eval("NewsSummary") %>
  </div>
  <div class="Clearer"> </div>
</div>
<div class="NewsBody">
  <%# Eval("NewsText") %>
</div>

<div class="NewsBody">
<cms:DocumentAttachments ID="ucDocAttachments" runat="server" TransformationName
="cms.root.attachment" Path='<%# Eval("NodeAliasPath") %>' />
</div>
```

```
</div>
```

Click **Save**. Click **OK** in the web part properties window.

4. If you go to the live site now and view some of the news, you should see the attachments displayed below the news text, just as in the screenshot below.



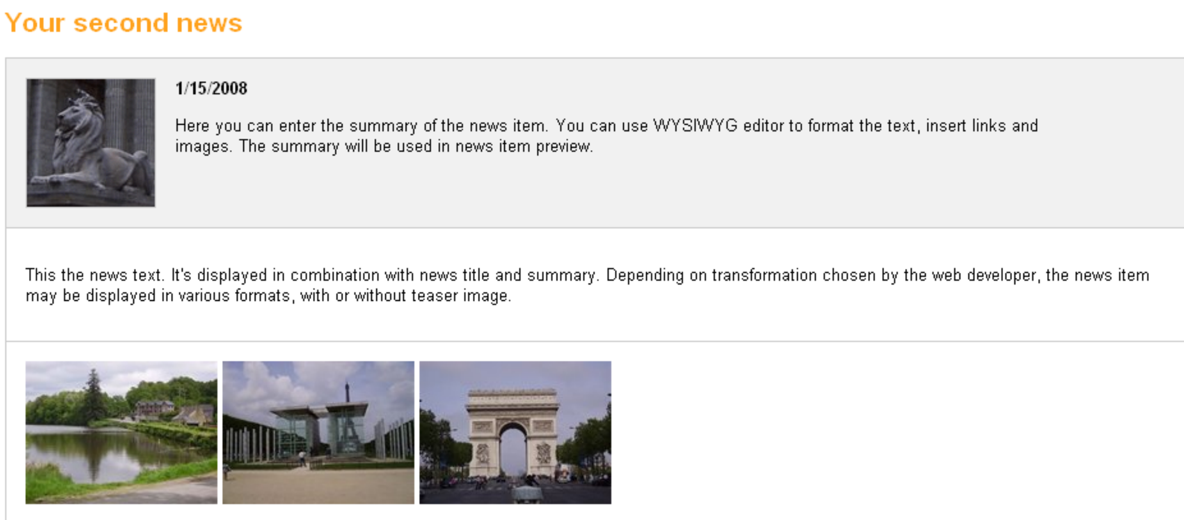
5. Let's try the other control now. Choose to **Configure** ( ) the **NewsRepeater** again and choose to **Edit** its **Selected item transformation**. Replace the transformation with the code below. Again, it is the original transformation with the highlighted parts added.

```
<%@ Register Src="~/CMSInlineControls/AttachmentLightboxGallery.ascx" TagName
="AttachmentLightboxGallery" TagPrefix="cms" %>

<div class="newsItemDetail">
<h1><%# Eval("NewsTitle") %></h1>
<div class="NewsSummary">
  <%# IfEmpty(Eval("NewsTeaser"), "", GetImage("NewsTeaser")) %>
  <div class="NewsContent">
    <div class="Date"><%# GetDateTime("NewsReleaseDate", "d") %></div>
    <%# Eval("NewsSummary") %>
  </div>
  <div class="Clearer"> </div>
</div>
<div class="NewsBody">
  <%# Eval("NewsText") %>
</div>

<div class="NewsBody">
```

```
<cms:AttachmentLightboxGallery ID="ucDocAttachments" runat="server"
TransformationName="cms.root.attachmentLightbox" SelectedItemTransformationName
="cms.root.attachmentLightboxDetail" Path='<%# Eval("NodeAliasPath") %>' />
</div>

</div>
```

Click **Save**. Click **OK** in the web part properties window.

6. If you go to the live site now, you should see the result as in the screenshot below. After clicking one of the images, it will be displayed in the lightbox.

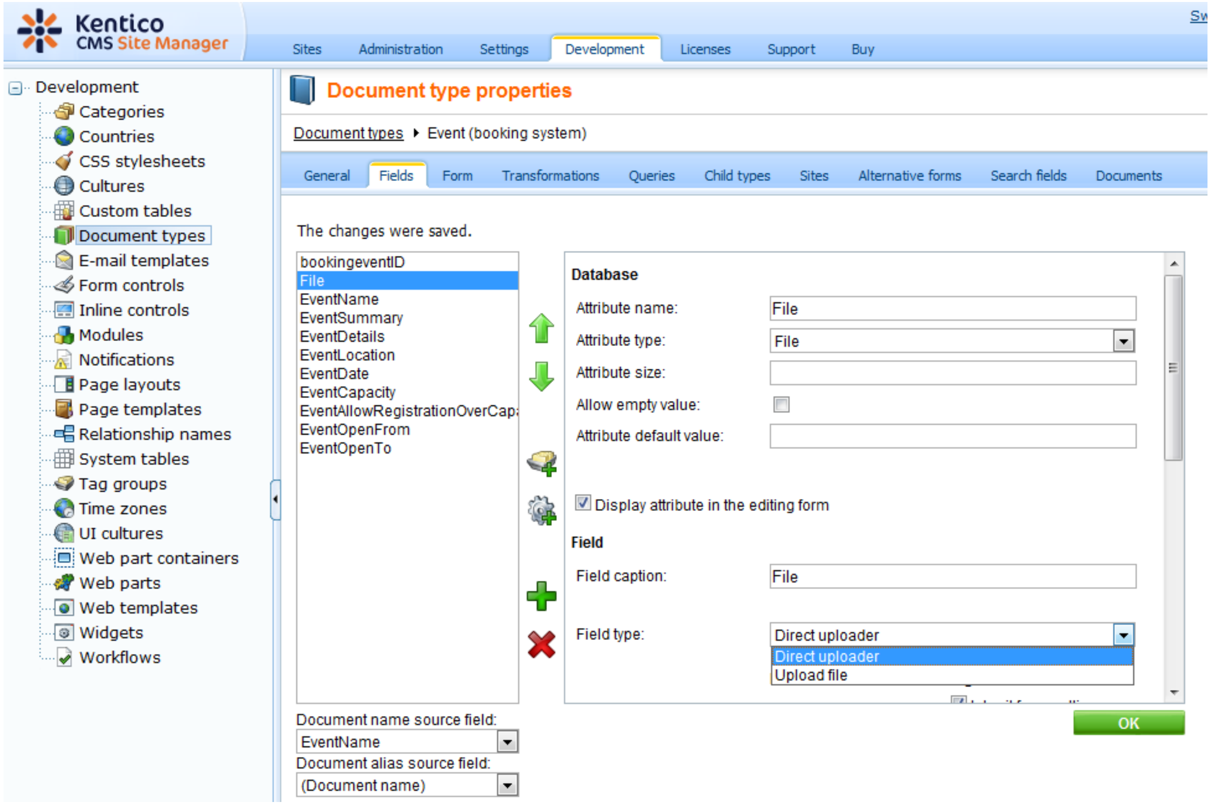


Please note: the appearance shown in this example is based on the controls' default transformations. You can fully customize the appearance by modifying the default transformations or creating your own transformations and specifying them in the **TransformationName** and **SelectedItemTransformationName** properties of the controls.

4.6.2.7 Using the File field

In the overview, we mentioned the **File** field used in the previous versions. Use of this field is still possible and could be particularly useful if you want to have a field where only one file can be uploaded. The default **CMS.File** document type is based on use of this type of field.

If you want to define a new **File** field, choose **File** from the **Attribute types** drop-down when defining a new field.



As you can see in the screenshot above, using the **Field type** drop-down, you can choose between two available controls for file upload which will be displayed on the **Form** tab:

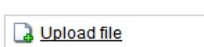
- ## Upload file

This is the original control which was used in the previous versions.

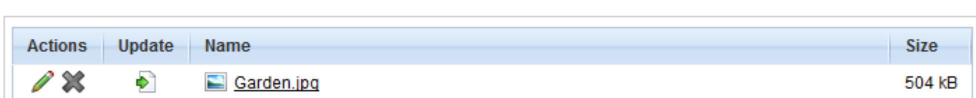
- ## Direct uploader

This is the same control that is used for uploading attachments via the **Form** tab. The only difference is that it allows only one file to be uploaded.

This is how it looks like when no file is uploaded:



And here is the control after file upload:

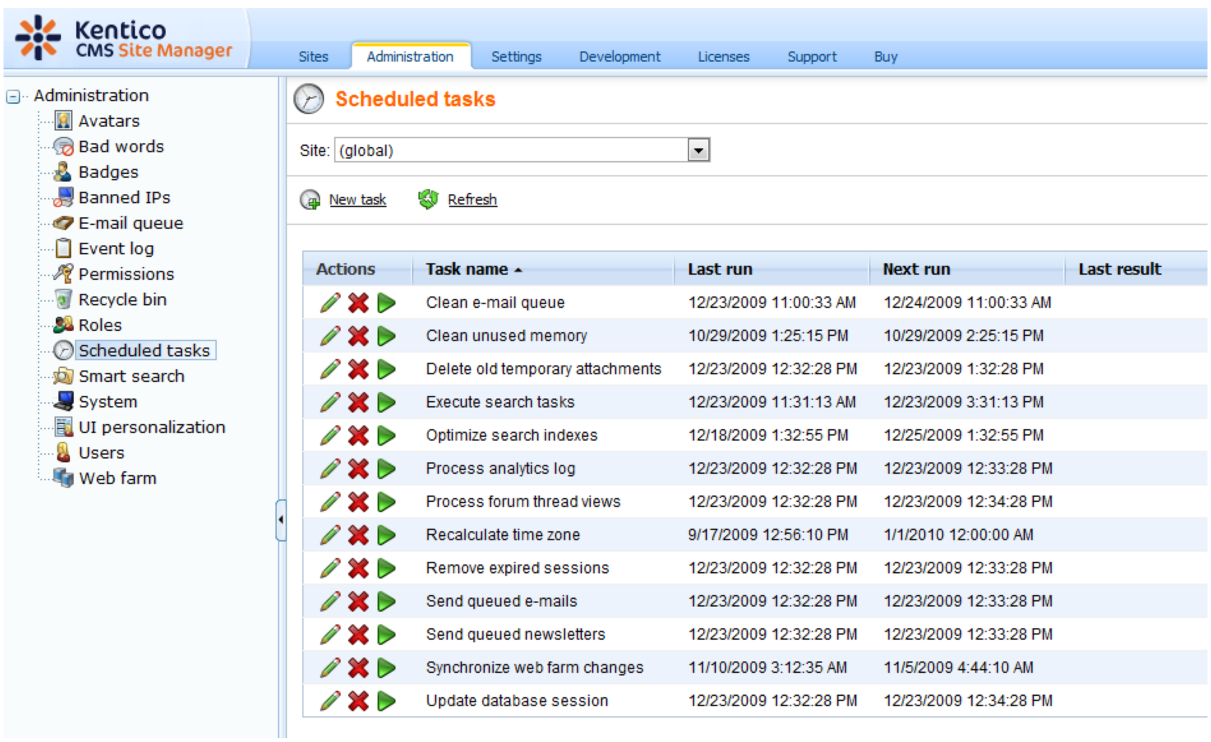


4.6.2.8 Temporary attachments handling

If you create a new document and start attaching files to it, temporary attachments are created. When the document is saved, the temporary attachment becomes a real attachment. If the document is not saved, the temporary attachments remain on the disk or in the database. To handle the unused files, there is a scheduled task pre-defined.

In **Site Manager -> Administration -> Scheduled tasks**, you can find the **Delete old temporary attachments** scheduled task. By default, this task is performed on a daily basis and deletes all temporary attachments older than 24 hours.

You can customize the interval by adding the <add key="CMSDeleteTemporaryAttachmentsOlderThan" value="1"/> key into the <appSettings> section of your site's *web.config* file. The value specifies the interval in hours.



4.6.2.9 Attachment names

Attachment names are unique for a document (or its version). If you try to upload an attachment that has the same name as some already existing attachment of the particular document, the new attachment gets a number in the **-(1)** format attached to its name.

This happens for the attachments to be accessible via friendly URLs in the following format:

- **~/getattachment/<node_alias_path>/<safe_filename>.<extension>**

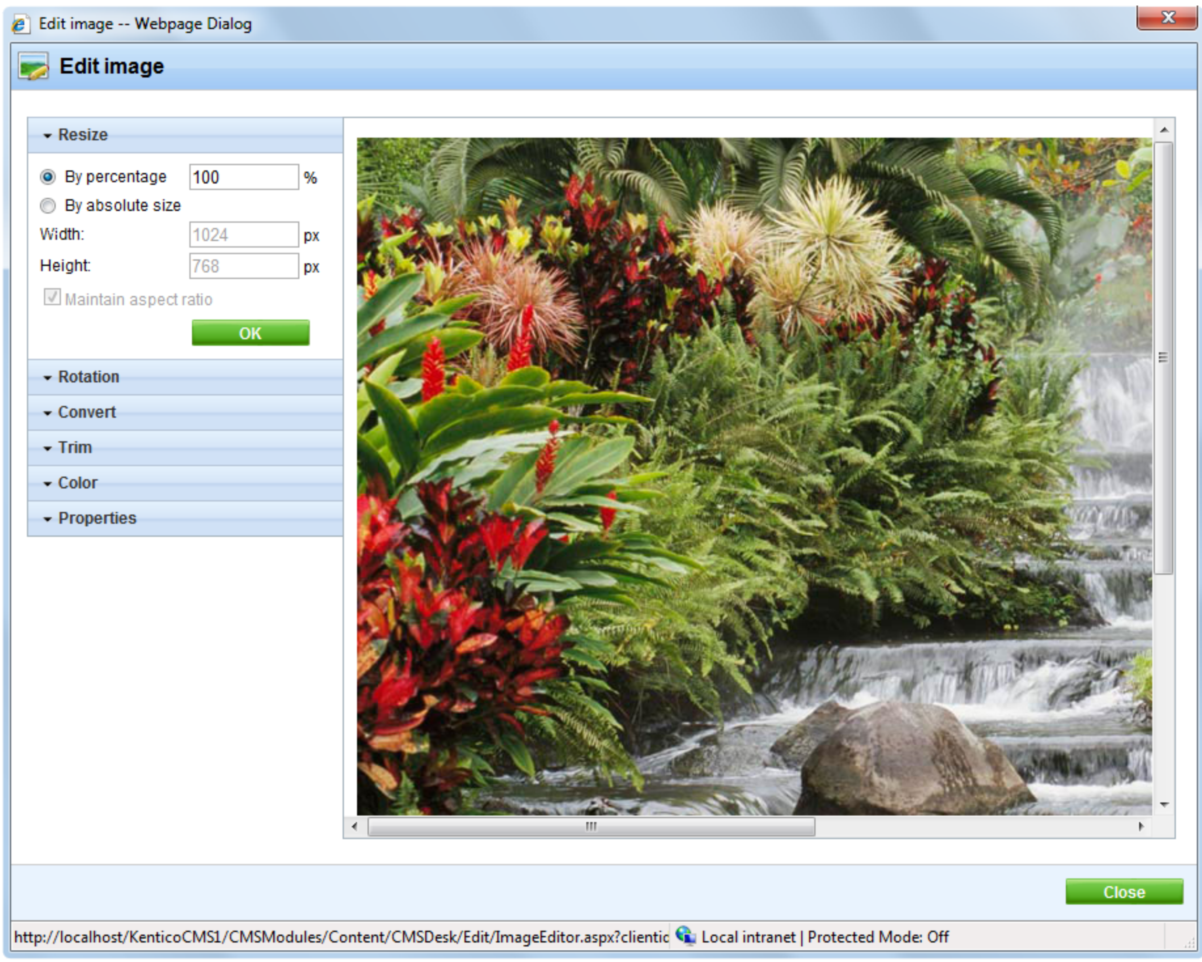
4.6.3 Handling images

4.6.3.1 Image editor

The image editor can be accessed throughout the whole system, everywhere the **Edit** ( ) icon is available for a listed image, just like:

- **document's Form tab**
- **media libraries**
- **document attachments**
- **etc.**

The image editor looks as in the following screenshot:



You can see the current appearance of the image on the right, while the following modifications can be done to the image on the particular tabs:

<u>**Resize**</u>

- **By percentage** - resizes the image to the entered percentage of side size
- **By absolute size** - resizes the image to the size specified in the **Width** and **Height** fields
- **Maintain aspect ratio** - if enabled, the other dimension is recalculated automatically when one dimension is specified above

Rotation

- **Rotate 90° left**
- **Rotate 90° right**
- **Flip horizontally**
- **Flip vertically**

Convert

- **From** - current format of the image
- **To** - target format of the conversion; **.bmp**, **.gif**, **.jpg**, **.png**
- **Quality** - quality of compression; applicable only for **.jpg** conversion

Trim

This action reduces the size of the image by removing the borders of the image.

- **Width** - width of the trimmed image
- **Height** - height of the trimmed image

Select the position of the trimming area in the original image and click the **Trim** button.

Color

- **Convert to grayscale** - if clicked, the image is converted from color to greyscale

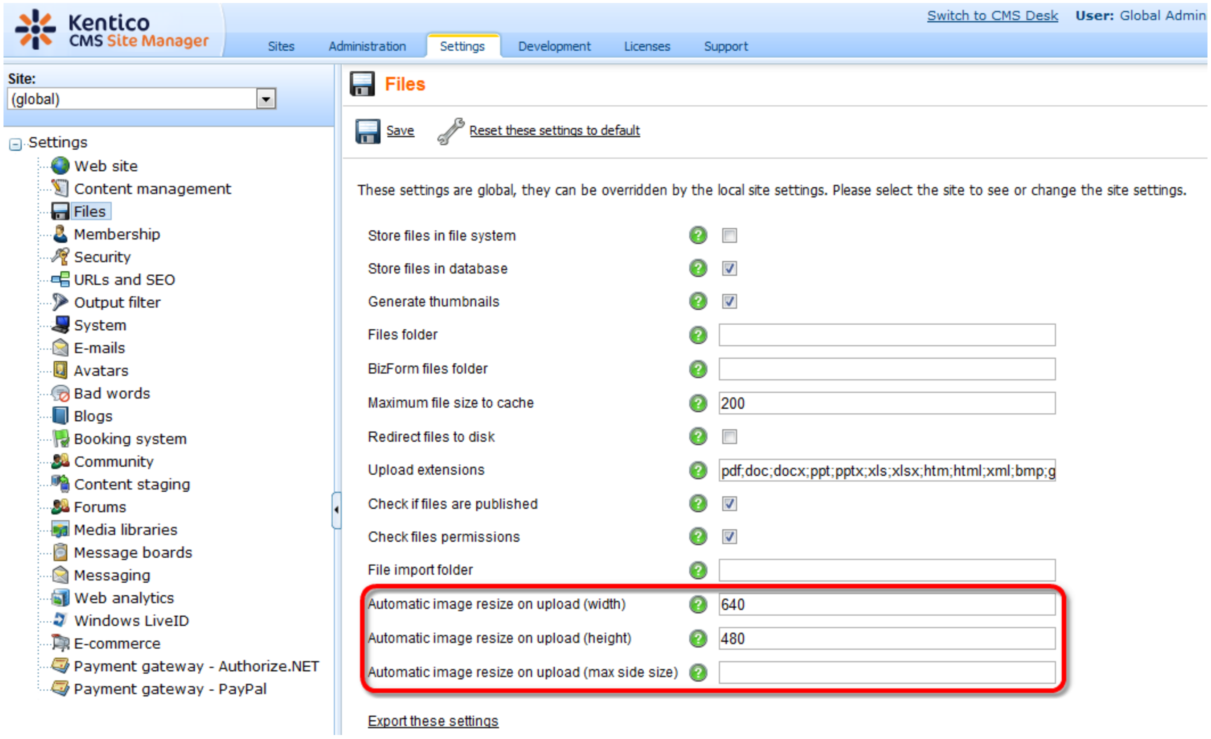
Properties

On this tab, you can rename the file using the **File name** field and you can also see the **Extension**, **Size**, **Width** and **Height** of the file.

4.6.3.2 Resizing images on upload

You can set global image resizing values in **Site Manager -> Settings -> Files** by entering the following values:

- **Automatic image resize on upload (width, height, max side size)** - depending on which values you fill in, the functionality is the following when uploading images:
  - **No values are entered** - images will not be resized
  - **Only width or only height** - images will be resized so that the width/height matches the entered value; the other dimension is also resized so that the aspect ratio is kept
  - **Both width and height** - images will be resized so that both dimensions match the entered values; the aspect ratio is not kept in this case
  - **Max side size** - if one of the sides of the image is larger than this value, the image will be resized so that the dimension of its larger side matches the entered value; the aspect ratio is kept and the width and height settings are not applied



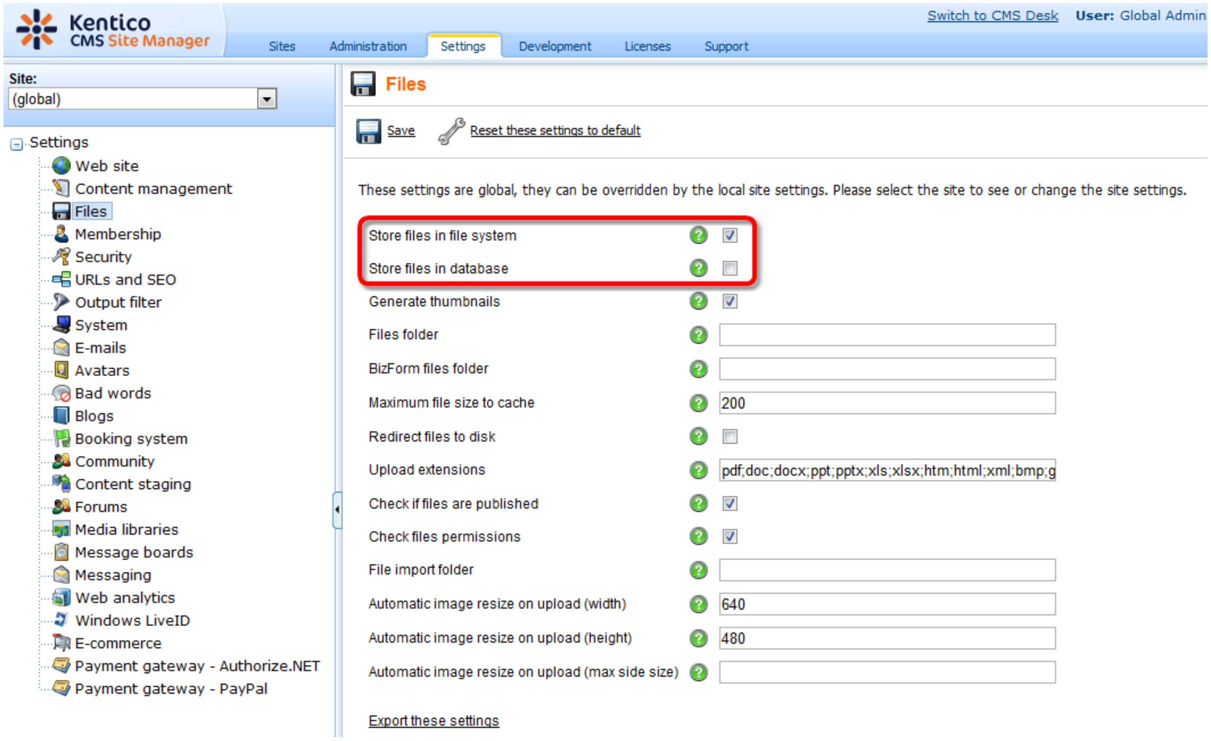
These settings are applied by default when uploading images as:

- **Document attachments**
- **Media library files**
- **CMS.File documents**
- **Editable text web part -** uploading images using the WYSIWYG editor dialogs
- **Editable image web part** - uploading images using the Select image dialog
- **Field editor fields -** uploading images using the following field types e.g. in BizForms, Document type field editor, etc.
  - **File** attribute type -> **Upload file** and **Direct uploader** field types
  - **Text** attribute type -> **HTML Area(Formatted Fext)**, **BBcode editor**, **Image selection**, **Media selection** and **File selection** field types

The default settings defined here can be **overridden by local settings** in the particular parts of the user interface (e.g. web part properties, field editor, WYSIWYG editor dialogs, etc.).

4.6.4 Where the files are stored

Document attachments and CMS. File documents can be physically stored in the file system, in the database or in both. You can define this in **Site Manager -> Settings -> Files**, using the **Store files in file system** and **Store files in database** options.



The following three combination can be achieved using the settings:

- **File system** - the files are stored in the configured folder on your disk. This option provides the best performance. However, the **Modify** permissions on the disk must be granted to the ASP.NET account on your machine, which is not always possible. The process of granting the Modify permission is described in this chapter.

- **Database** - the files are stored in the database. This option provides worse performance, but it allows you to use full-text search in uploaded files. It also doesn't require the Modify permission on the disk and it allows you to easily backup the uploaded files as a part of your database backup.

- **Both - Database and file system** - this option combines the advantages of both options. It provides the same performance as the file system-only option since the files are stored on the file system. At the same time, you can use the full-text search because you the database is also available.

Document attachments and metafiles location

Document attachments are stored in the file system under *~/<site code name>/files*, metafiles are stored under *~/<site code name>/metafiles*. The location of these folders can be customized in **Site Manager -> Settings -> Files -> Files folder**.

BizForm files location

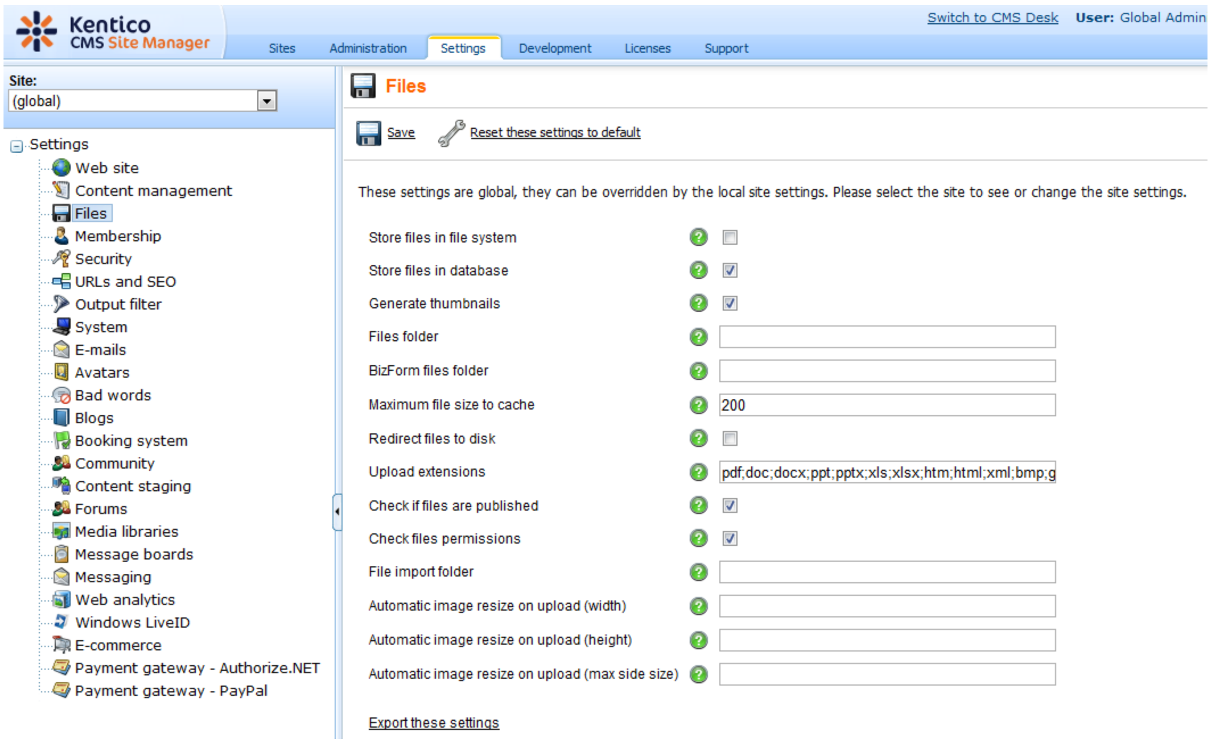
Files uploaded by site users into BizForms are always stored in the file system. The default location is *~/<site code name>/BizFormFiles*. You can customize the location in **Site Manager -> Settings -> Files -> BizForm files folder**.

Media library files location

Files stored in media libraries are always stored in the file system. The default location is *~/<site code name>/media*, while the location of the folder can be customized in **Site Manager -> Settings -> Media libraries -> Media libraries folder**, as described here.

4.6.5 Files-related settings

You can configure the file storage in **Site Manager -> Settings -> Files**.



The following options can be set on the page:

Store files in file system	Indicates if files should be stored in the file system.
Store files in database	Indicates if files should be stored in the database.
Generate thumbnails	Indicates if the CMS generates image thumbnails on the disk when a **resized version** of the image is displayed. This option only applies if files are stored in the file system. It **improves site performance**.
Files folder	The folder on the disk where document attachments and metafiles are stored. You can use: • **physical disk path** - e.g. c:\myfiles\mysite • **virtual path** - ~/UploadedFiles • **UNC path** - \\server\folder If you do not specify any value, the files are stored in folder **~/<site code**

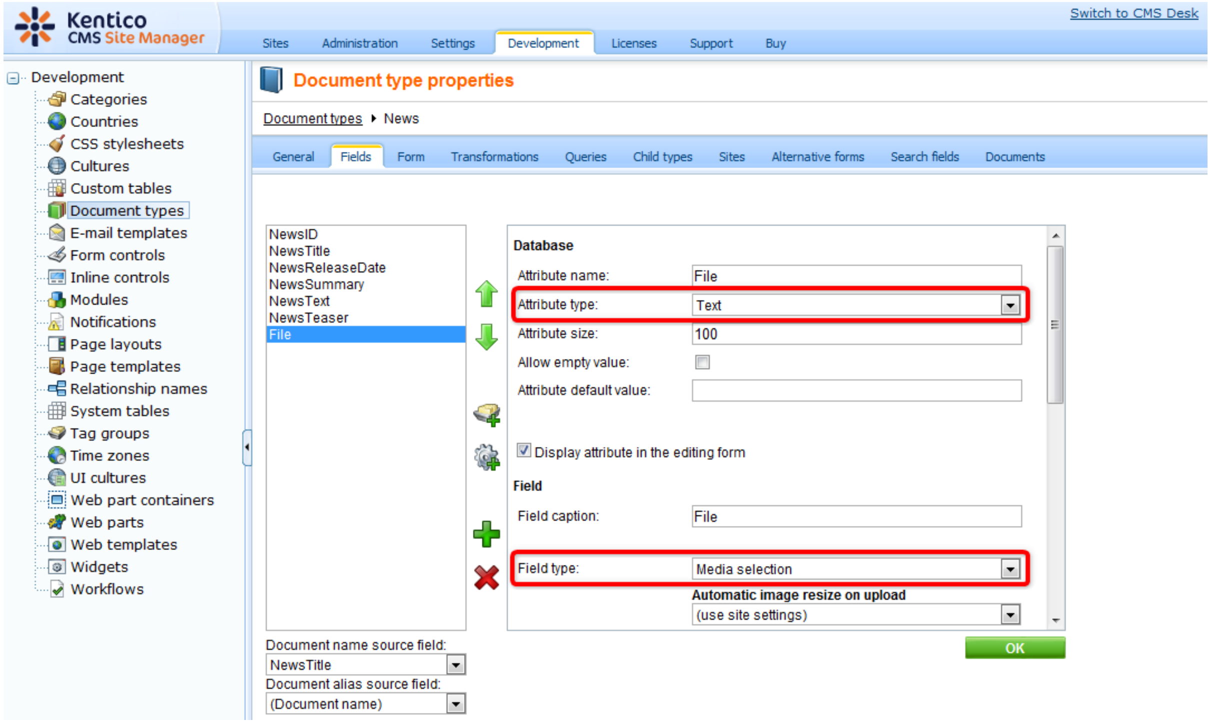
	name>/files.
BizForm files folder	Folder where files uploaded via BizForms are stored. If no value is entered, BizForm files are stored in *~/<site code name>/BizFormFiles*.
Maximum file size to cache	Specifies the maximum size of a file in kilobytes that is allowed to be cached.
Redirect files to disk	If checked, file requests are redirected to the physical file in the file system if possible.
Upload extensions	Allowed file extensions for the uploaded files. You can restrict the types of uploaded documents to (e.g. 'gif;jpg;doc;pdf'). It allows you to block users from uploading potentially dangerous files, such as ASPX scripts. If no value is specified, all types of files can be uploaded.
Check if files are published	If checked, only files that are in the **Published** workflow step can be accessed from the live site when a workflow is applied to the document.
Check files permissions	If checked, document permissions are applied to the files.
File import folder	Path to the source folder where files to be imported by the File import should be uploaded before import. If no value is entered, *~/CMSImportFiles* is used by default.
Automatic image resize on upload (width, height, max side size)	Depending on which values you fill in, the functionality is the following when uploading images: • **No values are entered** - images will not be resized • **Only width or only height** - images will be resized so that the width/height matches the entered value; the other dimension is also resized so that the aspect ratio is kept • **Both width and height** - images will be resized so that both dimensions match the entered values; the aspect ratio is not kept in this case • **Max side size** - if one of the image's sides is larger than this value, the image will be resized so that its larger side's dimension matches the entered value; the aspect ratio is kept and width and height settings are not applied More info can be found in Handling images -> Resizing images on upload.

4.6.6 Using the Media selection control

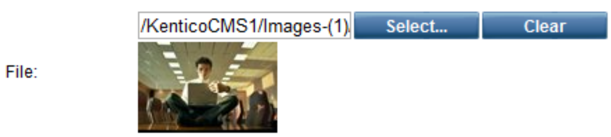
The Media selection form control can be used to enable users to select any types of files on a document's **Form** tab.

When defining the filed in field editor, you need to choose:

- **Attribute type**: Text
- **Field type**: Media selection

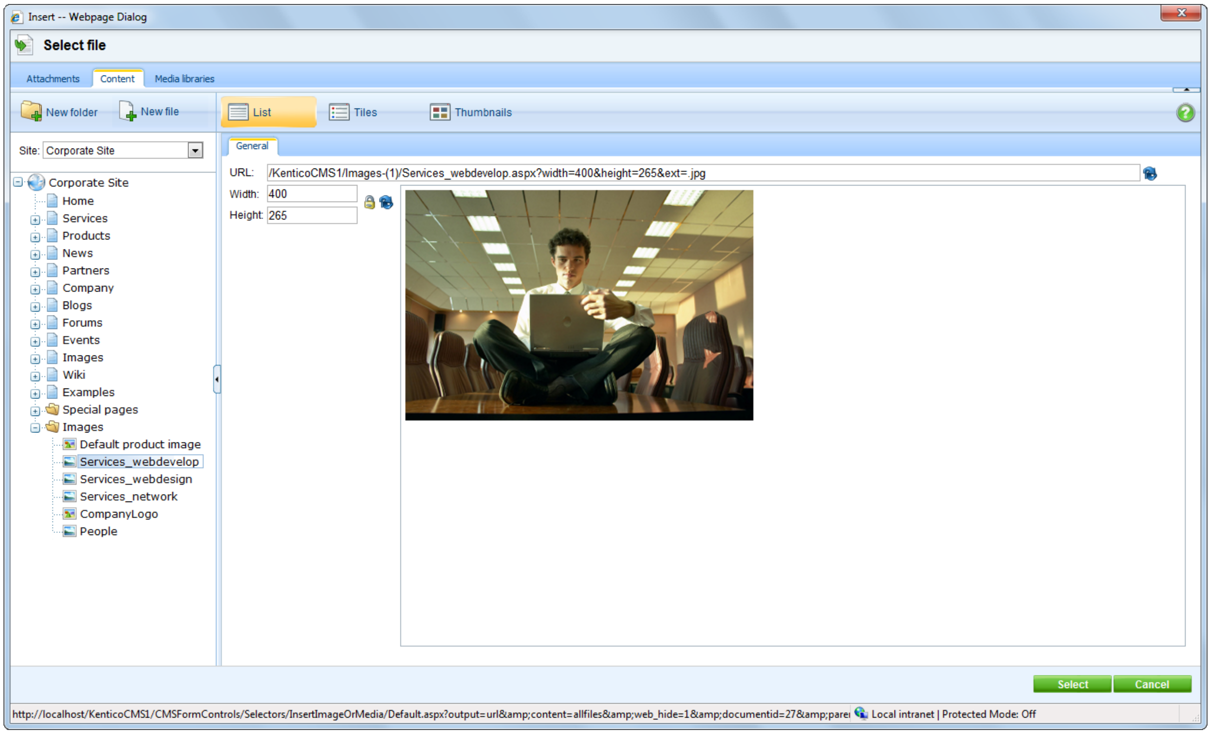


In the screenshot below, you can see the default appearance of the control on the **Form** tab. Notice that a preview of the currently selected image is displayed below the selection controls:



When the **Select...** button is clicked, the **Select file** dialog opens.

The dialog is similar to the Insert image or media dialog used in the WYSIWYG editor, while only the **Attachments**, **Content** and **Media libraries** tabs are available here. Description of the tabs can be found in the alike named parts of this page.



When you select a file, relative URL of the selected file is inserted into the field, with the with the **width**, **height** and **ext** parameters appended.

```
/KenticoCMS_0729/Images-(1)/Services_webdevelop.aspx?width=200&height=132&ext=.jpg
```

Displaying selected items in transformations

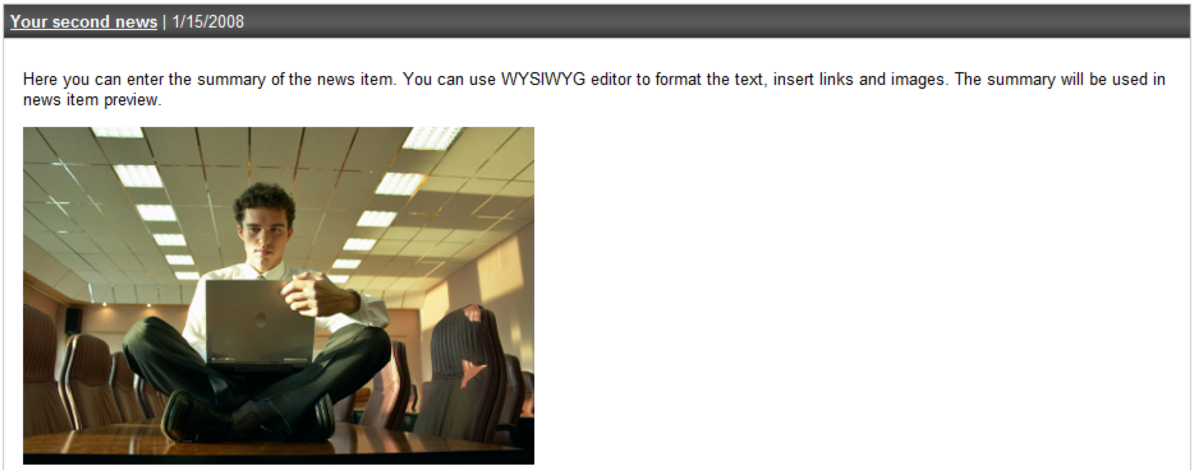
In transformations, items selected in the Media selection field can be displayed using the **~/ CMSInlineControls/MediaControl.ascx** inline control. When displaying the items in transformations, the **ext** parameter (extension of the file) defines what needs to be displayed (image, video player, flash player, ...).

The following code sample is a modified **CMS.News.NewsWithSummary** transformation with the control added (highlighted code). As you can see, you only need to register the MediaControl and then use it in the transformation just by providing it the URL. In this case, the URL is got

```
<%@ Register Src="~/CMSInlineControls/MediaControl.ascx" TagName="Media"
TagPrefix="cms" %>
<div class="NewsPTitle"><a href="<%# GetDocumentUrl() %>">
  <%# Eval("NewsTitle") %></a> | <%# GetDateTime("NewsReleaseDate",
"d") %>
</div>
<div class="NewsPBody">
  <%# Eval("NewsSummary") %>
  <cms:Media id="mediaElem1" runat="server" Url='<%# Eval<string>("File")%>' />
```

```
</div>
```

And this is the result rendered on the live site:



4.7 Document listing

On the **CMS Desk -> Content -> List** tab, you can see an **overview of documents placed under the currently selected document** in the content tree. You can filter the displayed pages using the filter above the list.

The **Name** column displays names of the documents in the currently edited culture. If the document's version in this culture **does not exist**, the column displays the name from the **default culture** with the default culture code appended in brackets.

The **Languages** column displays the translation statuses of the documents' particular language versions. More information about translation statuses can be found here.

The following operations can be performed with particular documents:

- **Edit** - by clicking the icon, you get redirected to the document's **Edit -> Page** tab in the currently selected culture
- **Delete** - deletes the document
- by clicking a document's name, the list tab will display the documents found under the clicked document

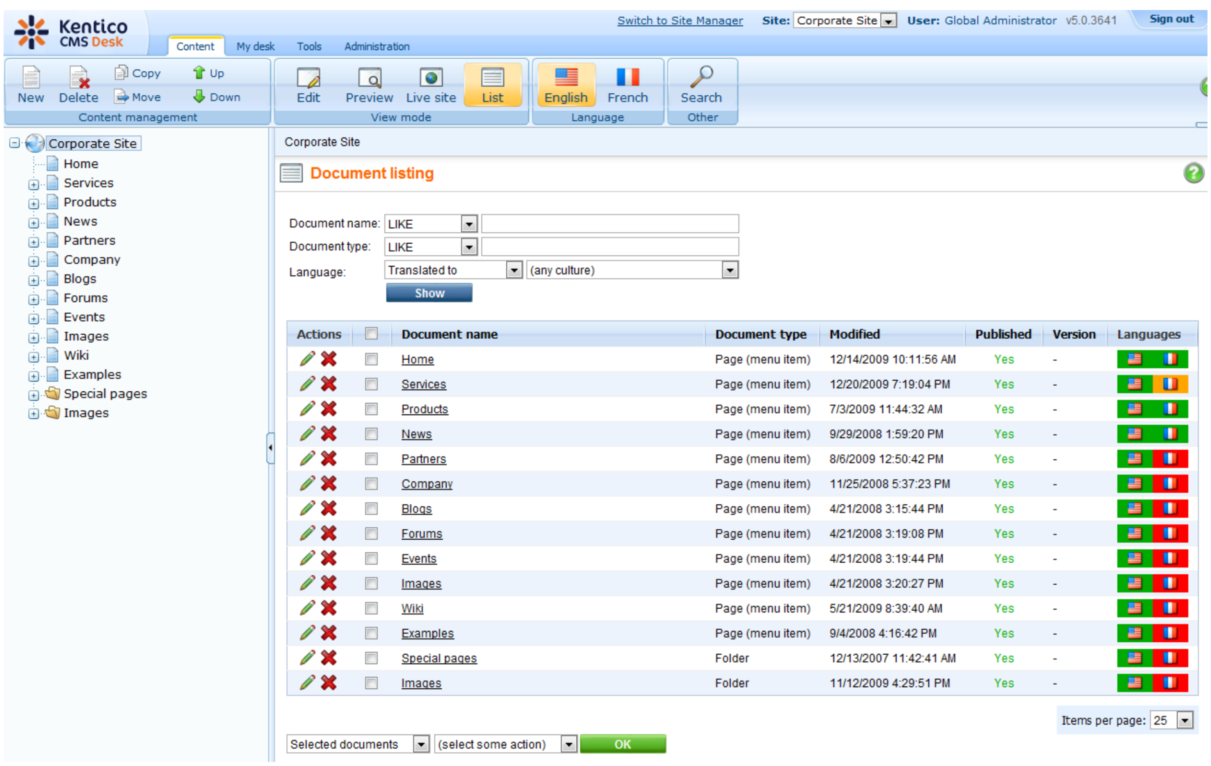
You can also perform **bulk actions** with the listed documents using the two drop-down lists at the bottom. First, you need to select from the following two options in the first drop-down list:

- **Selected documents** - performs the action only with the documents selected by the check-boxes ( )
- **All documents** - performs the action with all listed documents

Then you need to choose the action:

- **Move** - moves the documents to the location specified in a raised pop-up dialog
- **Copy** - copies the documents to the location specified in a raised pop-up dialog
- **Link** - creates a linked document in the location specified in a raised pop-up dialog
- **Delete** - deletes the documents
  - **Delete documents and their history (documents can not be restored)** - deletes the document permanently, i.e. without storing it in the recycle bin
  - **Delete all culture version of the specified documents** - only for multilingual sites; if enabled, all culture versions of the documents will be deleted; if disabled, only the versions in the currently edited culture will be deleted
- **Publish/Archive** - this option is available only for global administrators and publishes/archives the selected documents; before performed, a dialog gets displayed, listing the selected documents and offering the following extra options:
  - **Publish/Archive all culture versions of specified documents** - if enabled, all culture versions of the documents will be published/archived
  - **Publish/Archive also all child documents** - if enabled, all documents located under the selected documents will be published/archived too
  - **Perform Undo check-out for checked out documents** - this option is displayed only if you want to perform the operation for a document which is checked out (currently edited by some user); if you enable this options, such documents will be published/archived too

and click **OK** to perform the action.



4.8 Content search

The interface for searching the documents in the content tree can be accessed by clicking the **Search** () icon in **CMS Desk -> Content**.

1. First, you need to decide if you want to perform the search using Smart search or SQL search. This

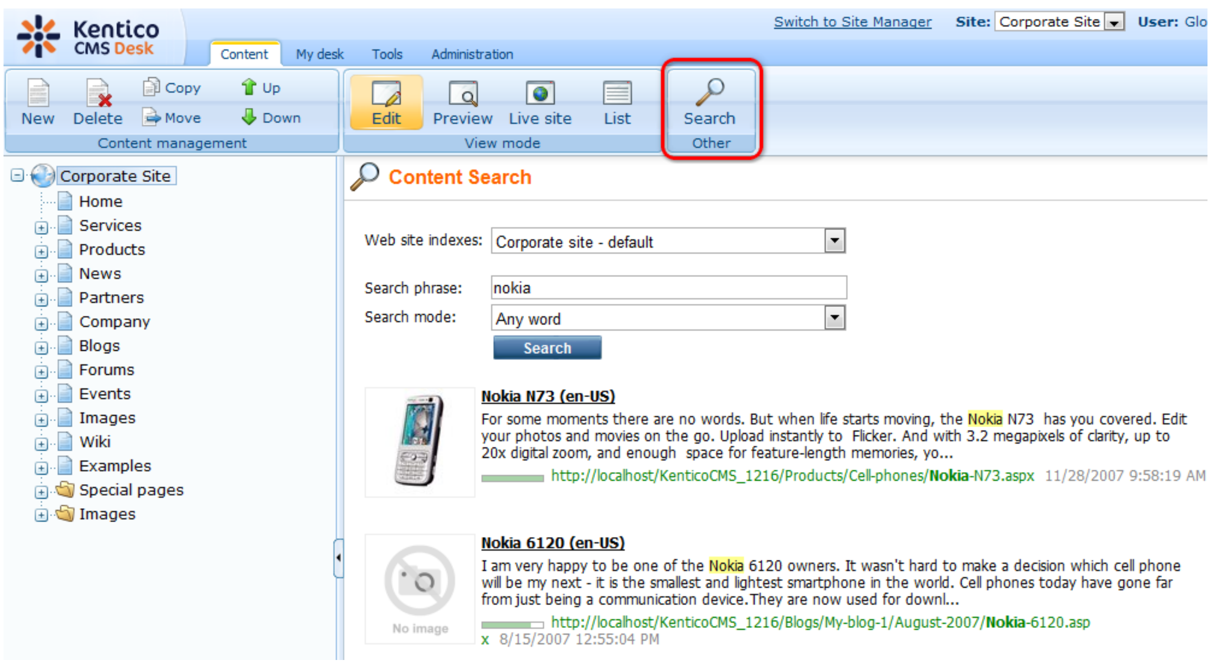
depends on the selection made in the **website indexes** drop-down:

- **SQL search** - you need to select *(SQL Search - default)*; slower, but supports search in both published and unpublished documents
- **Smart search** - you need to select a particular smart search index; fast, but does not support search in unpublished documents

2. Then you need to specify the following criteria:

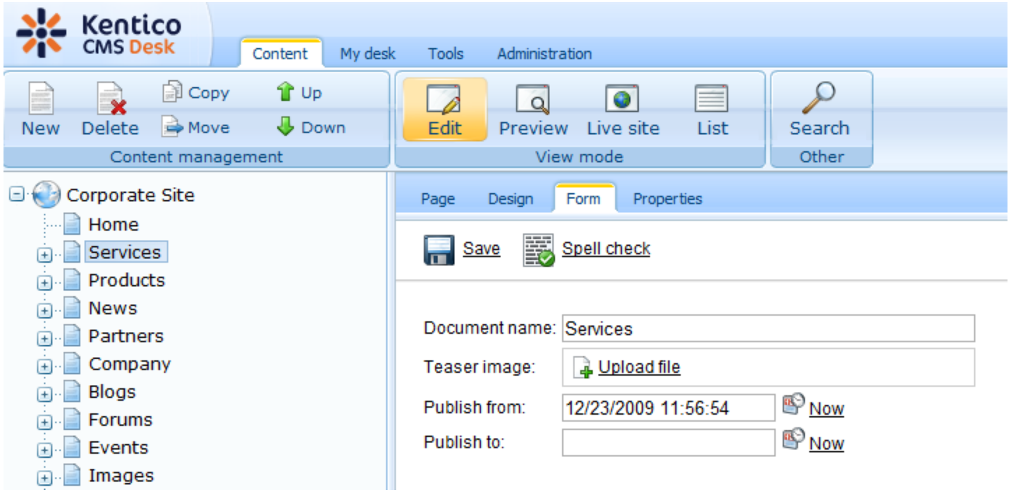
- **Search phrase**: the text that you are looking for; if you are searching using a Smart search index, you can use the syntax described here; if you are searching using SQL search, standard SQL syntax can be used
- **Search mode**: specifies how the search phrase will be used:
  - *Exact phrase* - returns documents where the search phrase is found exactly as entered
  - *Any word* - returns documents where at least one word of the search phrase is found
  - *All words* - returns documents where all words of the search phrase are found, regardless of their position or order in the text

Click **Search**. Found documents will be listed as in the screenshot below.



4.9 Content scheduling

Kentico CMS allows you to specify when the document will be published. When you edit the document form, you can typically find the Publish from/Publish to fields at the end of the form:



When you set these values, the document will be displayed on the website only in the given time period.

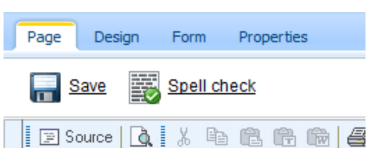
If you do not set the **Publish from** value, the document will be displayed on the live site immediately. If you do not set the **Publish to** value, the document will never expire.

Content scheduling and workflow/versioning

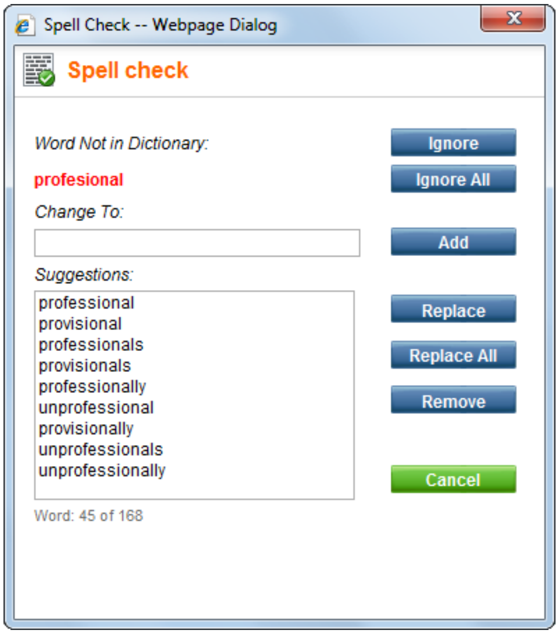
If you set publish from/to values to documents that use workflow, they will not be published before they are approved. However, the publishing time may not be exact since the publishing is ensured by a scheduled process that is executed every minute by default. You can check the status of this process in **CMS Desk -> Administration -> Scheduled tasks** -> choose the website and search for the **Content scheduling** task.

4.10 Using the built-in spell-checker

You can spell-check all the content on the **Page** and **Form** tabs using the built-in spell-checker:



When you click the **Spell check** button, the spell-checker reads all text fields and checks their content. If it finds any typo, it shows the dialog like this:



You can then ignore the word, add it to the dictionary or replace it with suggested word.

Please note: If you add a new word to dictionary it's only saved in the current session. The next time you sign in to Kentico CMS, the added words will be lost.

Dictionaries

The dictionary is used based on the currently chosen content culture. If no dictionary is available for the current content culture, the default dictionary is used. The default dictionary is specified in the **CMSDefaultSpellCheckerCulture** configuration key in the **appSettings** section of the **web.config** file. By default, it's set to en-US.

Adding addtional dictionaries

The dictionaries are stored in folder **<web project>\App_Data\Dictionaries**. If you need some additional dictionaries, you can download them from the following URLs:

- AR-ae: http://www.kentico.com/Downloads/SpellChecker/ar-AE.zip
- CS-cz: http://www.kentico.com/Downloads/SpellChecker/cs-cz.zip
- DE-de: http://www.kentico.com/Downloads/SpellChecker/de-de.zip
- EL-gr: http://www.kentico.com/Downloads/SpellChecker/el-gr.zip
- EN-au: http://www.kentico.com/Downloads/SpellChecker/en-au.zip
- EN-ca: http://www.kentico.com/Downloads/SpellChecker/en-ca.zip
- EN-gb: http://www.kentico.com/Downloads/SpellChecker/en-gb.zip
- EN-us: http://www.kentico.com/Downloads/SpellChecker/en-us.zip
- ES-es: http://www.kentico.com/Downloads/SpellChecker/es-es.zip
- ES-mx: http://www.kentico.com/Downloads/SpellChecker/es-mx.zip
- FR-fr: http://www.kentico.com/Downloads/SpellChecker/fr-fr.zip
- HE-il: http://www.kentico.com/Downloads/SpellChecker/he-il.zip
- IT-it: http://www.kentico.com/Downloads/SpellChecker/it-it.zip
- NB-no: http://www.kentico.com/Downloads/SpellChecker/nb-no.zip
- NL-nl: http://www.kentico.com/Downloads/SpellChecker/nl-nl.zip

- NN-no: http://www.kentico.com/Downloads/SpellChecker/nn-no.zip
- PL-pl: http://www.kentico.com/Downloads/SpellChecker/pl-pl.zip
- PT-pt: http://www.kentico.com/Downloads/SpellChecker/pt-pt.zip
- RO-ro: http://www.kentico.com/Downloads/SpellChecker/ro-ro.zip
- RU-ru: http://www.kentico.com/Downloads/SpellChecker/ru-ru.zip
- TH-th: http://www.kentico.com/Downloads/SpellChecker/th-th.zip

and unpack them to the dic folder. Then, you should restart the website using **Site Manager -> Administration -> System dialog -> click Restart application**. The file name of the dictionary must match the culture code of the currently edited content - e.g. fr-fr.

4.11 Accessing content using .NET code (API)

You can make all changes to the content programmatically using Kentico CMS API. It allows you to script any actions (create a new document, update document, etc.) using .NET code.

You can find more details and examples in chapter Content management internals.

4.12 Permissions and security

Kentico CMS allows you to configure permissions for particular site sections or even particular documents. You can find more details in the Membership and security chapter.

4.13 FAQ

This topic provides information about the most common issues and their solutions. If you do not find the answer here, please contact our technical support.

Q: The CMS Desk doesn't display the content tree or the WYSIWYG editor doesn't open.
A: Please check that you use one of the supported browsers and that you have JavaScript enabled. You should also unblock the pop-up blocker or any similar blocker for the website with Kentico CMS.

Q: I have modified the document, but the changes are not displayed on the live site.
A: This may be caused by several reasons:

1. **Caching** - if caching is used, the changes may not be displayed on the website immediately.
2. **Workflow** - if you use workflow, the changes are published on the live website only after the document has been approved in all workflow steps.

Q: I have defined related documents, but they are not displayed anywhere on the page.
A: You (your developer) need to add some web part/control that will display the related documents - e.g. the **Listings and viewers/Related documents** web part or **Repeater** web part.

Q: I need to add custom field to the document. Is it possible?
A: Yes, every structured document type can be customized with your own fields. See Document types for details.

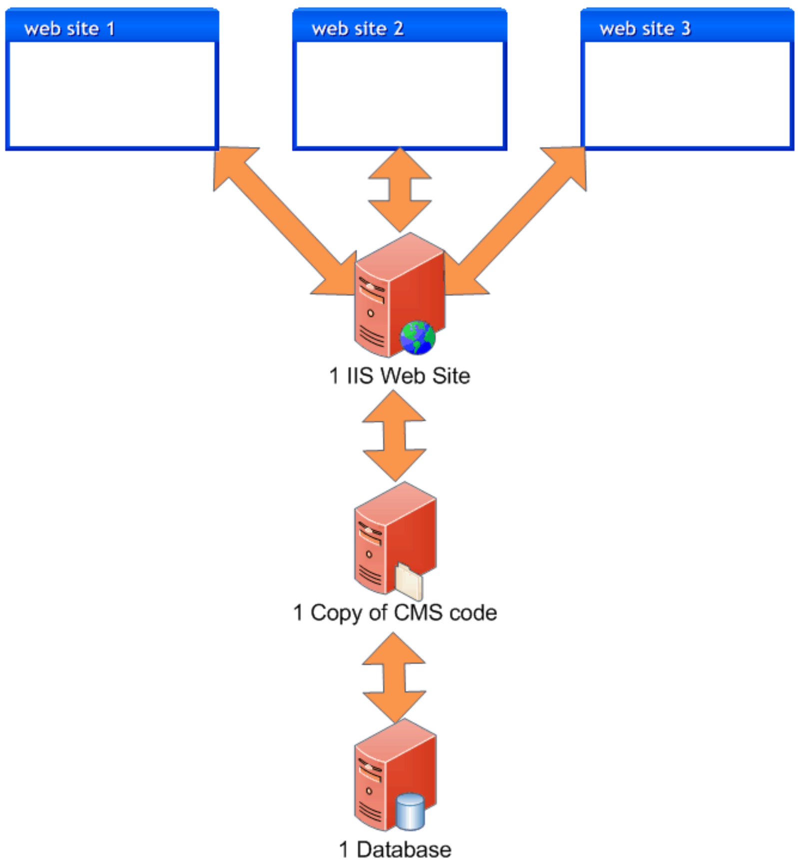
Part V

Managing sites

5 Managing sites

5.1 Site Management Overview

Kentico CMS allows you to manage multiple sites in a single installation. The database contains data for all websites and the websites are managed using a single administration interface (single copy of code). The following figure shows the multi-site configuration where one database and one copy of CMS code are used for multiple websites.



When you use multi-site configuration, you can share:

- documents
- users
- global settings and system tables
- document types
- page templates
- web parts

This feature is useful if you need to create multiple websites for a single company and share users/ documents/settings between them.

When to choose separate installations

There are situations in which we recommend running separate instances of Kentico CMS for every website:

- You build websites with **many documents** and performance is critical for you.
- Your customers have very **different requirements** and you need to customize some common parts of the system, such as administration interface or structure of shared tables.
- Your customers are very sensitive to **security** and you do not want to risk that some other client will get access to other websites by administrator's mistake.

5.2 Managing sites

You can manage the websites in **Site Manager -> Sites** dialog.
Creating a new website
Please see New site wizard for more details on creating a new website.

General tab

Site display name	The name of the site displayed to the users.
Site code name	The name of the site used in the code.
Site domain name	The main domain of the website. Use domain name without http:// protocol and without www. If you use other port than 80, specify it as well. Correct: mycompany.com partners.mycompany.com mycompany.com:8080 Incorrect: http://mycompany.com www.mycompany.com
Default content culture	Default culture of the site content. It can be changed using the **Change** button.
Default visitor culture	The content culture that should be set to site visitors if they don't have any cookie with preferred culture set. (Automatic) means the culture is decided based on the browser preferences. If none the preferred languages is supported by the current site, the default culture is used.
Site CSS stylesheet	Default CSS style sheet used for all pages unless they override the value with their own CSS stylesheet.
Editor CSS stylesheet	CSS style sheet used for the WYSIWYG editor content.
Site description	Optional description of the website for internal use.

Domain aliases tab

Here you can add domain aliases that point to this website.

Example of domain aliases

If your website uses **mycompany.com** as its Main domain and you also wish to use the **my-company.co.uk** domain and have it access the same website, you need to add the **my-company.co.uk** alias to the list of domain aliases.

Culture tab

Here you can choose in which languages the website content is available. You will use this tab to configure website that provides content in multiple languages.

5.3 Starting and stopping sites

You can run and stop websites using the **Start site** and **Stop site** buttons in the **Sites** dialog.

Switching between Sites on a Single Domain

If you try to run a site that uses the **same domain name (or alias)** as another site that is already running, you will get an error message and the site will not be started.

If you need to test **several websites on a single domain**, such as http://localhost, you need to specify the domain (localhost) for multiple websites and start only one of them.

If you cannot use your own domain names, you can use several alternatives that point to the same computer with different host name http://localhost, http://127.0.0.1 or http://mycomputer.

You can find more details on how to configure the websites in chapter Configuring multiple websites.

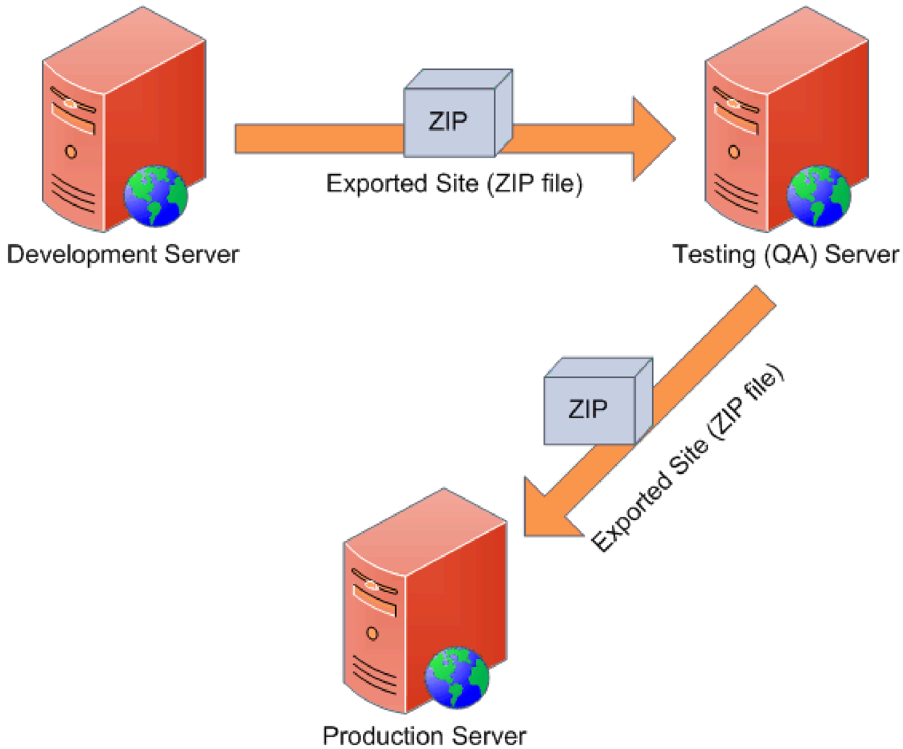
5.4 Creating a new site

Please see New site wizard.

5.5 Export and import

5.5.1 Overview

You can export and import website content and settings from one Kentico CMS instance to another. You can use this feature to move website or chosen objects between development, testing (QA) and production (live) server as illustrated in the figure below:

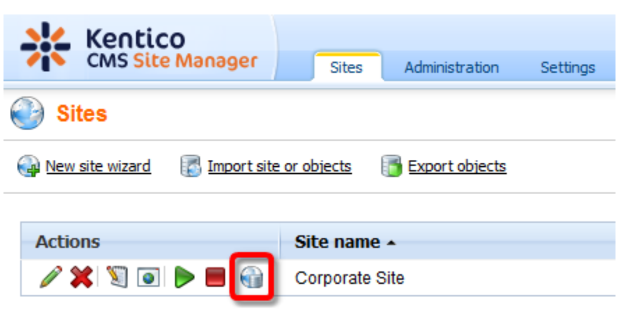


5.5.2 Exporting a site

Kentico CMS allows you to export whole website including its settings and related objects (such as document types, workflows, web parts, page templates, etc.) into a single file that can be imported on the same or different Kentico CMS instance.

Exporting a site

Go to **Site Manager -> Sites** and click the **Export site** icon next to the site you want to export. This will start the export wizard.

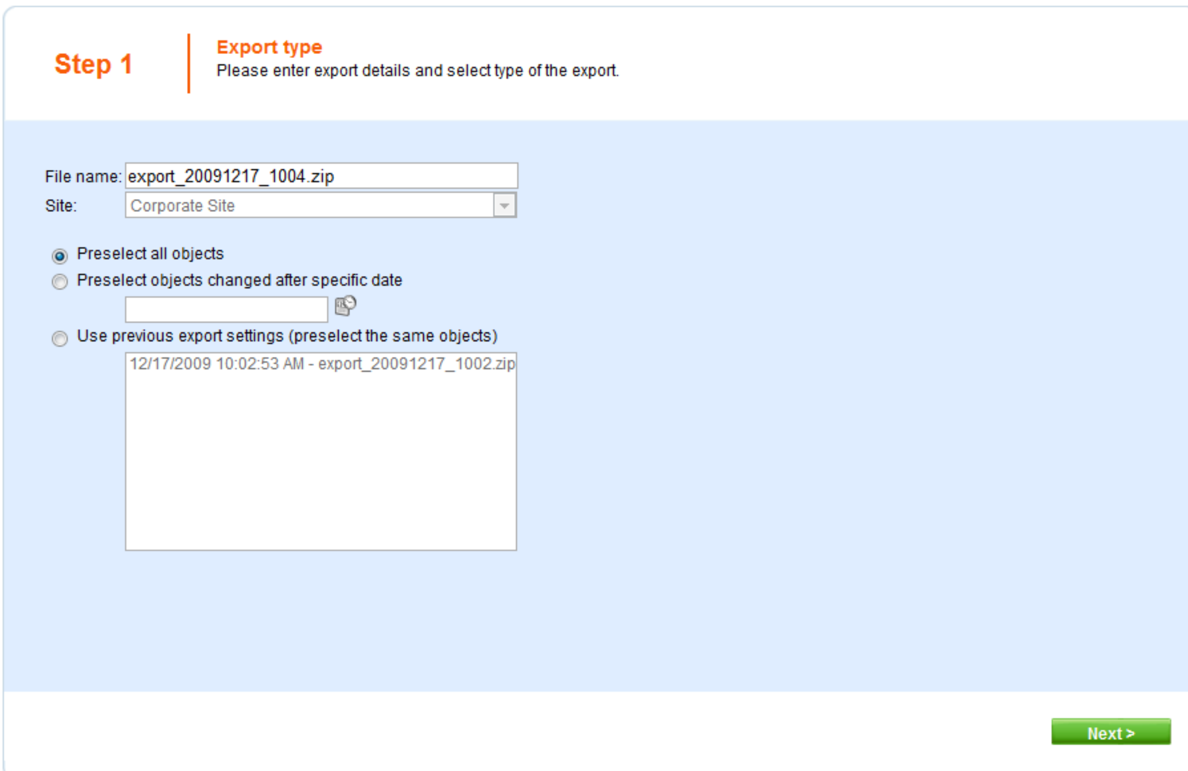


In the first step of the wizard, you have to fill in the name of the export package and choose type of objects pre-selection:

File name - name of the export package; a default name will be pre-filled and the package will be stored in **<web project>\CMSSiteUtils\Export**.

- **Preselect all objects** - all site objects will be preselected in the next step
- **Preselect objects changed after specific date** - only objects changed after the specified date will be selected in the next step
- **Use previous export settings** - settings used in a previous export selected from the list below will be used

Select the option that suits your purposes and click the **Next** button.



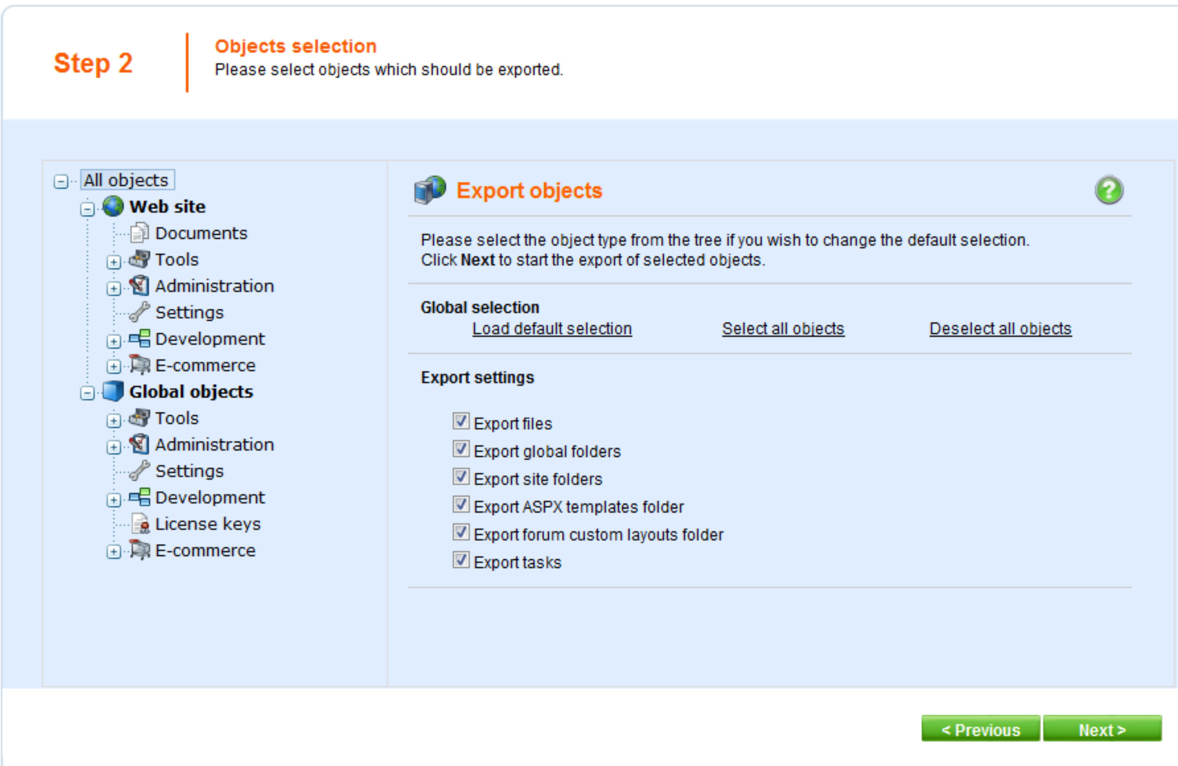
In **Step 2**, you can select which objects will be exported. The tree on the left represents object categories. These reflect the sections of the administration interface under that the objects can be found. The **Site** category contains objects related to the selected website. The **Global objects** category contains global objects that can be used by all sites. By selecting a category, a set of check boxes appears in the right part of the screen, letting you select which objects will be exported. If you select the root of the tree (All objects), you will be offered with the following options:

Global selection

- **Load default selection** - objects will be selected based on the pre-selection choice made in Step 1
- **Select all objects** - all objects will be selected
- **Deselect all objects** - no objects will be selected

Export settings

- **Export files** - some objects in the database are linked with physical files in the file system; if you check this check-box, these files will be exported along with the database objects
- **Export global folders** - if checked, global files under the folders listed below will be exported
  - *<web project>\App_Code\Global*
  - *<web project>\CMSGlobalFiles*
- **Export site folders** - if checked, files under the folders listed below will be exported
  - *<web project>\App_Code\<site code name>*
  - *<web project>\<site code name>*
- **Export ASPX templates folder** - if checked, folder with ASPX page templates will be exported
  - *<web project>\CMSTemplates*
- **Export forum custom layouts folder** - if checked, folder with custom forum layouts will be exported
  - *<web project>\CMSModules\Forums\Controls\Layouts\Custom*
- **Export tasks** - if checked, delete tasks (incremental deployment) will be included in the package



The following categories contain extra options to be set:

Documents

- **Export documents** - if checked, documents will be exported

- **Export document histories** - if checked, histories of all exported documents will be exported
- **Export document relationships** - if checked, relationships of all exported documents will be exported
- **Export document level permissions** - if checked, document security settings made in CMS Desk will be exported

- **Export blog comments** - if checked, blog comments will be exported

- **Export event attendees -** if checked, event attendees will be exported for all exported events

BizForms

- **Export BizForms data -** if checked, stored BizForms' data will be exported together with the exported BizForms

Forums

- **Export forum posts -** if checked, forum posts will be exported together with the exported forums

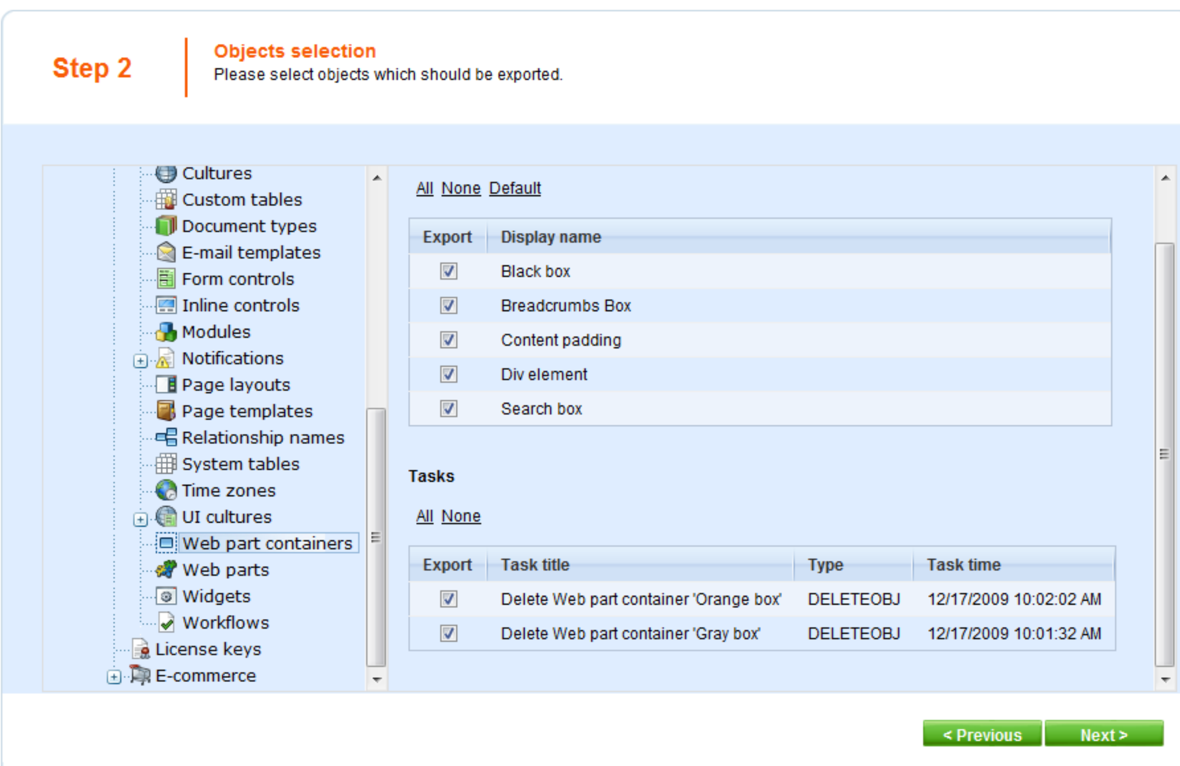
Message boards

- **Export board messages -** if checked, board messages will be exported together with particular message boards

Media libraries

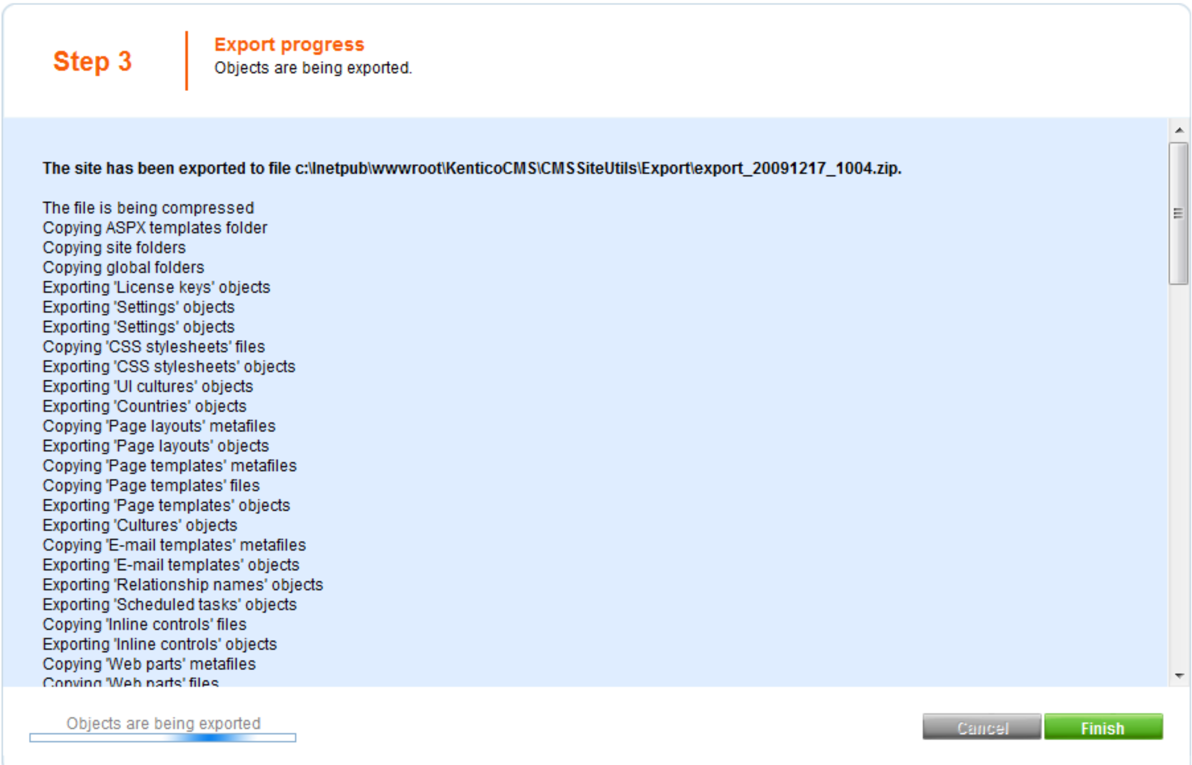
- **Export media files -** if checked, media files stored in the database will be exported
- **Export physical files -** if checked, physical media files stored in the file system will be exported; this option is not selected by default as it may cause the package size grow extremely large; instead, it is recommended to export these files manually

If you have the "**Log export tasks**" option enabled in **Site Manager -> Settings -> Content Staging**, a list of object deletion tasks may also be displayed at the bottom of the list. This happens when some objects have been deleted (just as the two Web part containers in the screenshot below). If you leave the check-boxes checked, the objects will be deleted after importing the package on the target server.



After making all required selections, click **Next** to proceed to the next step.

A log appears, showing you the progress of exporting. You can abort exporting by clicking the **Cancel** button at any time. When exporting finishes, a message appears at the top of the log, telling you the full path to the exported file. Click the **Finish** button. You will be redirected back to **Site manager** -> **Sites**
.



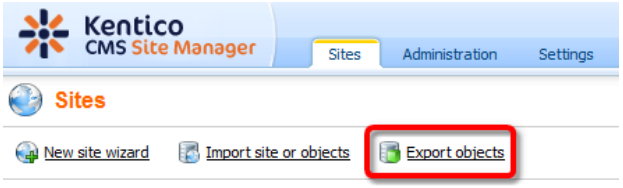
Now you can copy the exported package to the target installation of Kentico CMS, into the **<web project>\CMSSiteUtils\Import** folder and use the **Import site or objects** wizard described in the Importing a site or objects topic.

5.5.3 Exporting objects

Besides exporting the whole website, you can also choose to export only selected objects (web parts, document types, page templates, etc.). This is useful when you update some object on the development machine and want to copy the updated object to the staging or production server.

Exporting selected objects

Go to **Site Manager -> Sites** and click **Export objects**. The Export objects wizard starts.

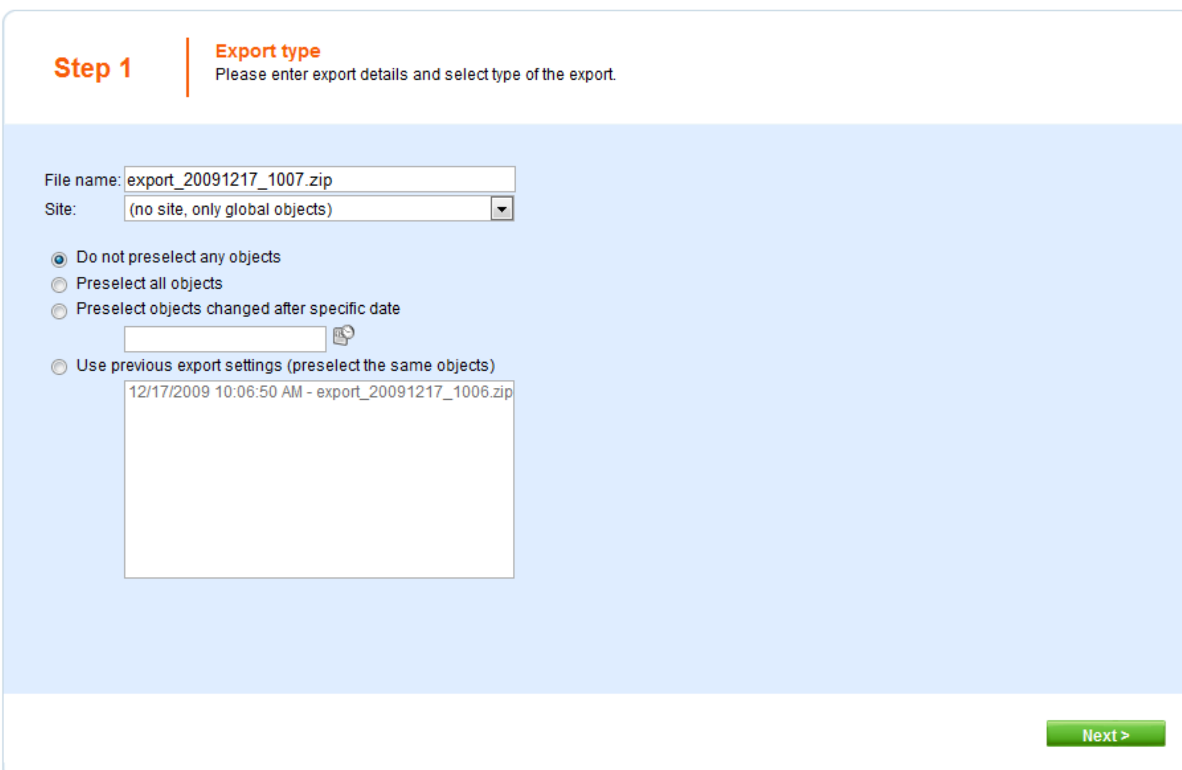


In the first step of the wizard, you have to fill in the name of the export package and choose type of objects pre-selection:

- **File name** - name of the export package; a default name will be pre-filled and the package will be stored in **<web project>\CMSSiteUtils\Export**.

- **Do not preselect any objects** - no objects will be preselected in the next step
- **Preselect all objects** - all objects will be preselected in the next step
- **Preselect objects changed after specific date** - only objects changed after the specified date will be preselected in the next step
- **Use previous export settings** - settings used in a previous export selected from the list below will be used

Select the option that suits your purposes and click the **Next** button.



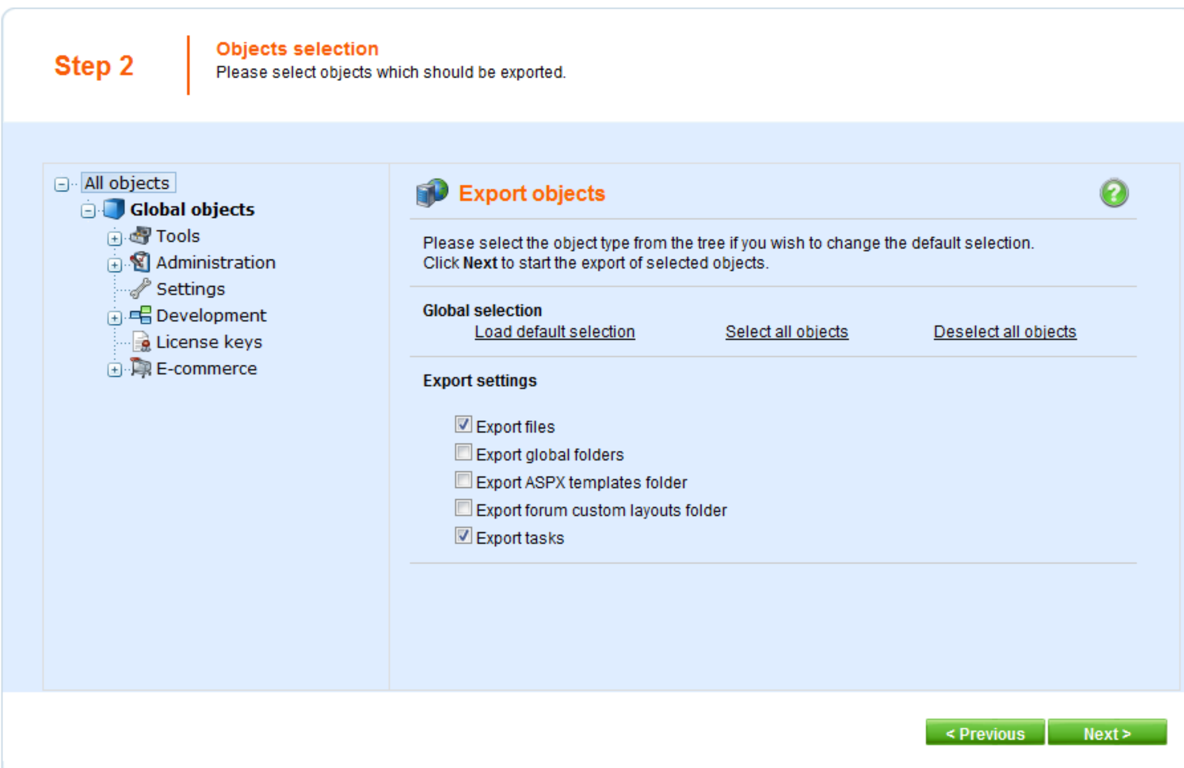
In **Step 2**, you can select which objects will be exported. The tree on the left represents object categories. These reflect the sections of the administration interface under that the objects can be found. The **Site** category contains objects related to the selected website. The **Global objects** category contains global objects that can be used by all sites. By selecting a category, a set of check boxes appears in the right part of the screen, letting you select which objects will be exported. If you select the root of the tree (All objects), you will be offered with the following options:

Global selection

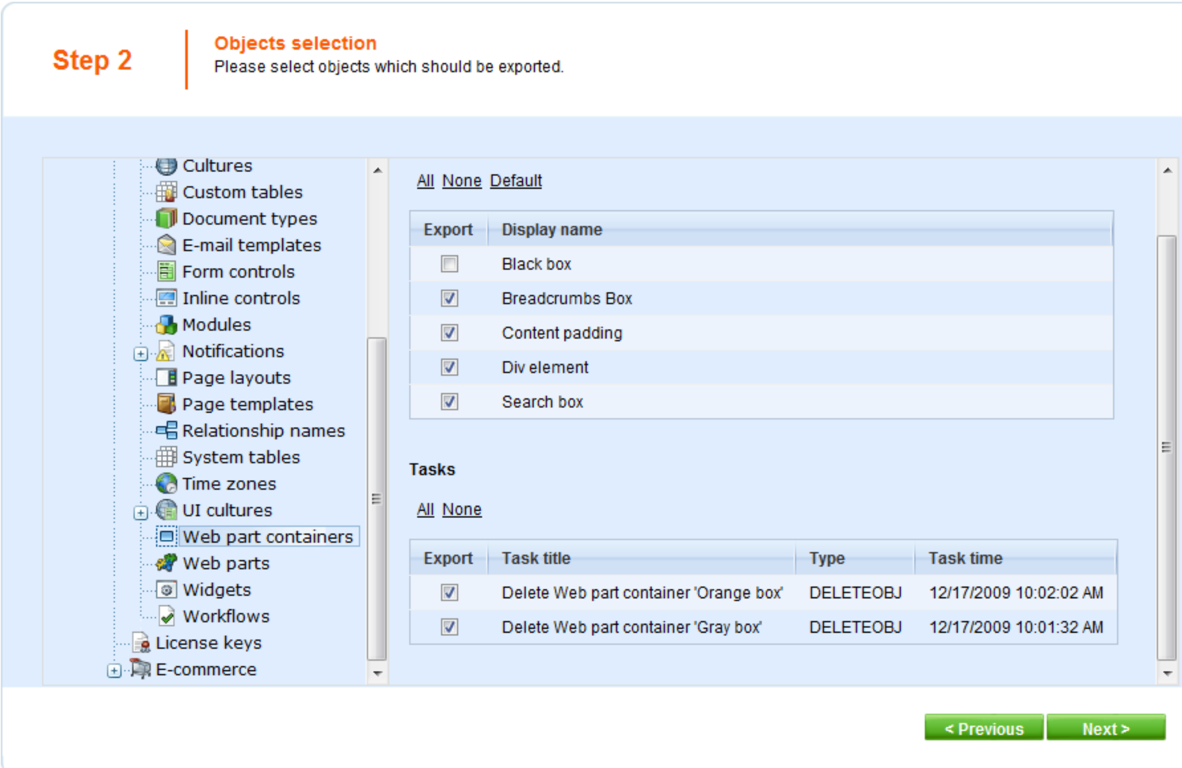
- **Load default selection** - objects will be selected based on the pre-selection choice made in Step 1
- **Select all objects** - all objects will be selected
- **Deselect all objects** - no objects will be selected

Export settings

- **Export files** - some objects in the database are linked with physical files in the file system; if you check this check-box, these files will be exported along with the database objects
- **Export global folders** - if checked, global files under the folders listed below will be exported
  - *<web project>\App_Code\Global*
  - *<web project>\CMSGlobalFiles*
- **Export ASPX templates folder** - if checked, folder with ASPX page templates will be exported
  - *<web project>\CMSTemplates*
- **Export forum custom layouts folder** - if checked, folder with custom forum layouts will be exported
  - *<web project>\CMSModules\Forums\Controls\Layouts\Custom*
- **Export tasks** - if checked, delete tasks (incremental deployment) will be included in the package

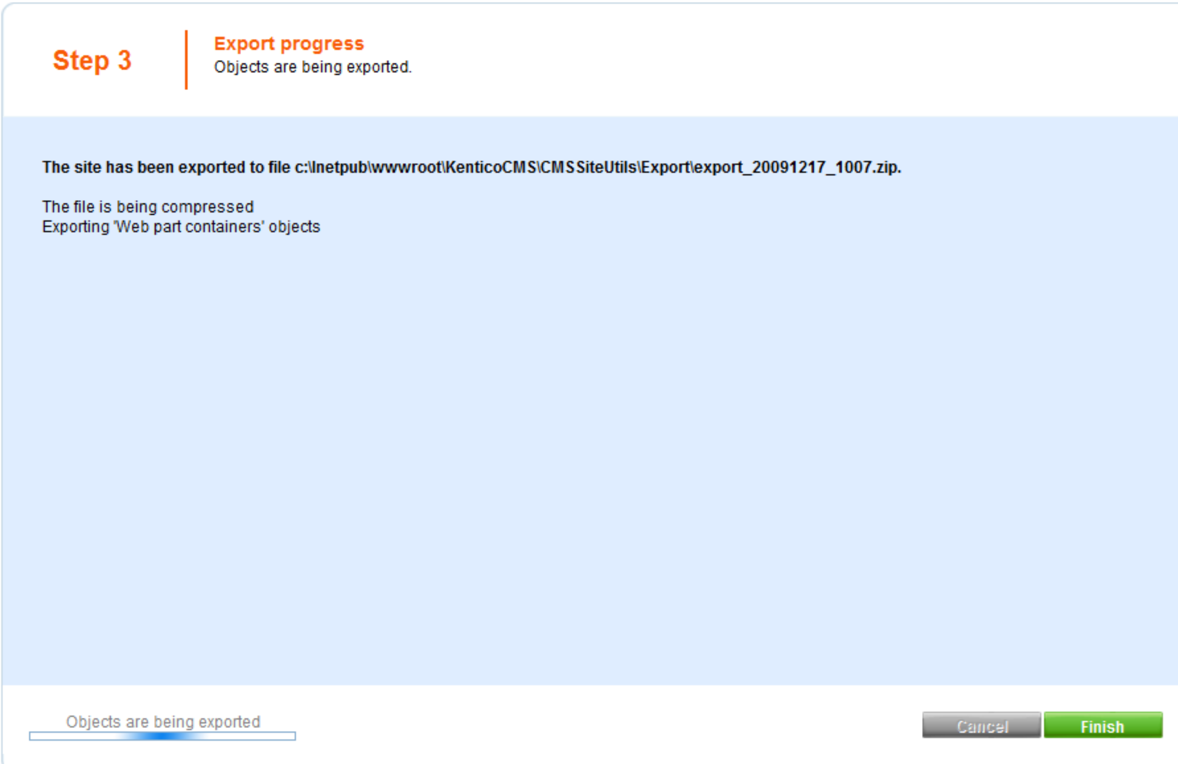


If you have the "**Log export tasks**" option enabled in **Site Manager -> Settings -> Content Staging**, a list of object deletion tasks may also be displayed at the bottom of the list. This happens when some objects have been deleted (just as the two Web part containers in the screenshot below). If you leave the check-boxes checked, the objects will be deleted after importing the package on the target server.



After making all required selections, click **Next** to proceed to the next step.

A log appears, showing you the progress of exporting. You can abort exporting by clicking the **Cancel** button at any time. When exporting finishes, a message appears at the top of the log, telling you the full path to the exported file. Click the **Finish** button. You will be redirected back to **Site manager** -> **Sites**.



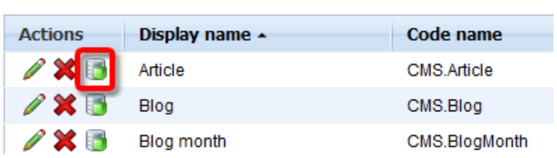
Now you can copy the exported file to the target installation of Kentico CMS, into the **<web project>\CMSSiteUtils\Import** folder and use the **Import site or objects** wizard described in the Importing a site or objects topic.

5.5.4 Exporting single objects

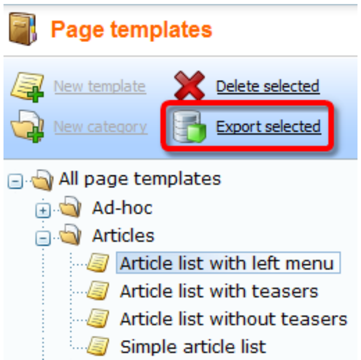
Some objects can also be exported separately as single object packages. This is useful when you want to quickly export only one or few objects apart from the rest of all other objects.

Single object export is supported for: CSS stylesheets, Document types, E-mail templates, Form controls, Inline controls, Page layouts, Page templates, Web part containers, Web parts, Workflow schemas.

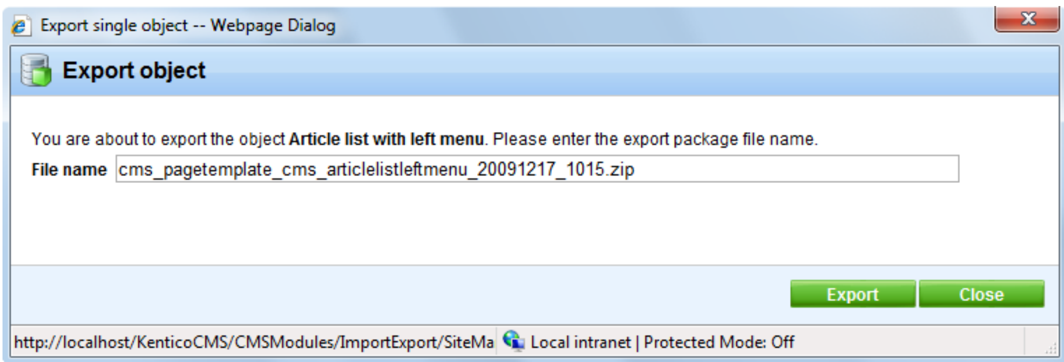
These objects can be found in the corresponding sections in **Site manager -> Development**. You can export an object by clicking the **Export object** ( ) icon.



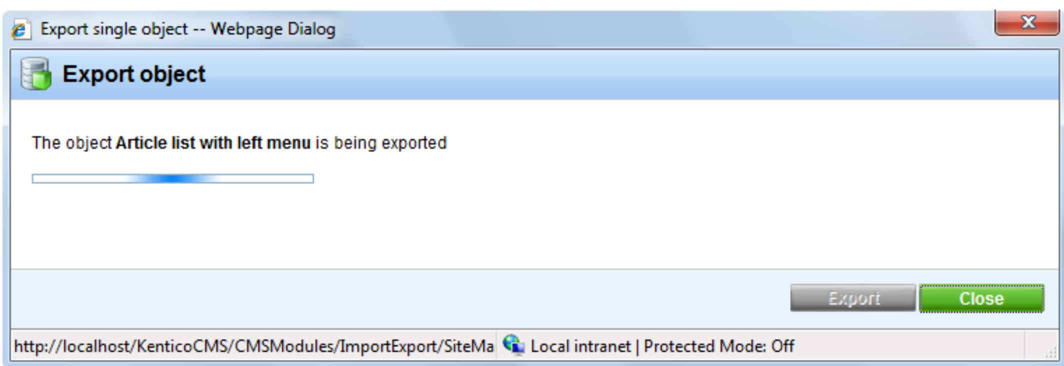
Or the **Export selected** link in case of Page templates and Web parts.



The following window will appear. Enter the name of the export file (default name will be pre-entered) and click **OK**.



Now wait while the object is being exported.



When the exporting is successfully finished, the following message with path to the exported file will be displayed. The **Download object** link below can be used for storing the file into a different location. Click it to open the typical file download dialog of your web browser.

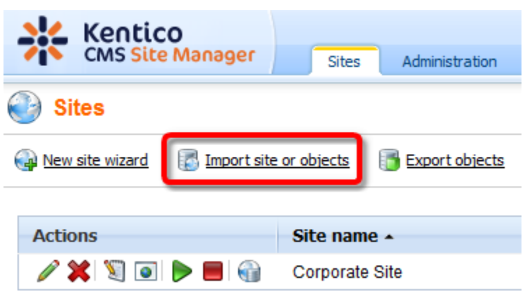
Finally, close the window by clicking the **Close** button.

5.5.5 Importing a site or objects

After you export some site or objects using the **Export site wizard** or **Export objects wizard**, you can import it using the **Import site or objects wizard**. Before you start the wizard, you need to copy your exported packages into the **<web project>\CMSSiteUtils\Import** folder.

Importing a site

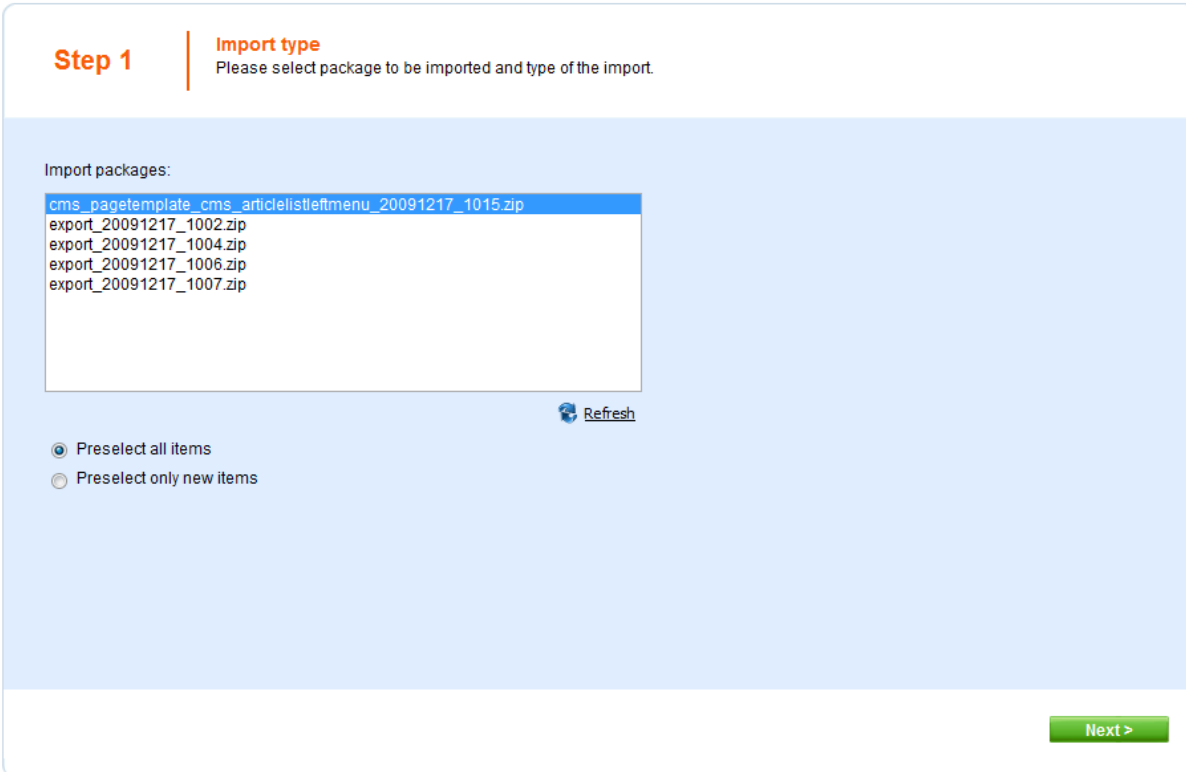
Go to **Site Manager -> Sites** and click **Import site or objects**.



In the first step, a list of **Import packages** located in *<web project>\CMSSiteUtils\Import* is displayed. If you want to add an import package to the folder, do it and click the **Refresh** ( ) link. The new package will be displayed in the list.

Using the two radio buttons below, you can determine if duplicate items will also be preselected or if only new items will be selected for import in the next step.

Choose to import the appropriate package and click the **Next** button.

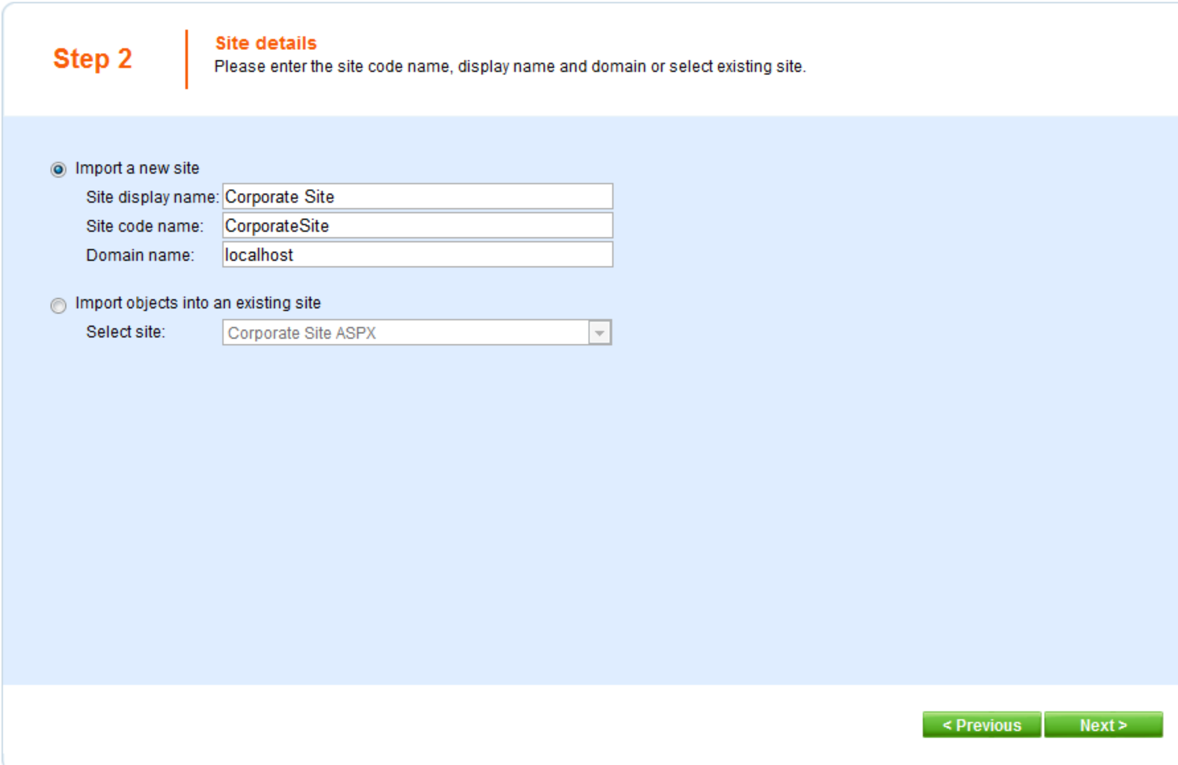


Step 2 will be displayed only when you are importing a site package. In case that you are importing a package containing only global objects, Step 3 will be displayed instead.

You have the following two options in Step 2:

- **Import a new site** - when chosen, a new site will be created based on the contents of the package; you have to enter the site's display name, code name and domain name
- **Import objects into an existing site** - when chosen, contents of the package will be imported into the site chosen by the check-box below

Click the **Next** button to proceed to the next step.



In **Step 3**, you can select which of the objects from the package will be imported. The tree on the left represents object categories. These reflect the sections of the administration interface under that the objects can be found. By selecting a category, a set of check boxes appears in the right part of the screen, letting you select which objects will be imported. Objects that already exist on the target server are marked with *. If you leave the check-box of such object checked, this existing object will be overwritten with the newly imported one.

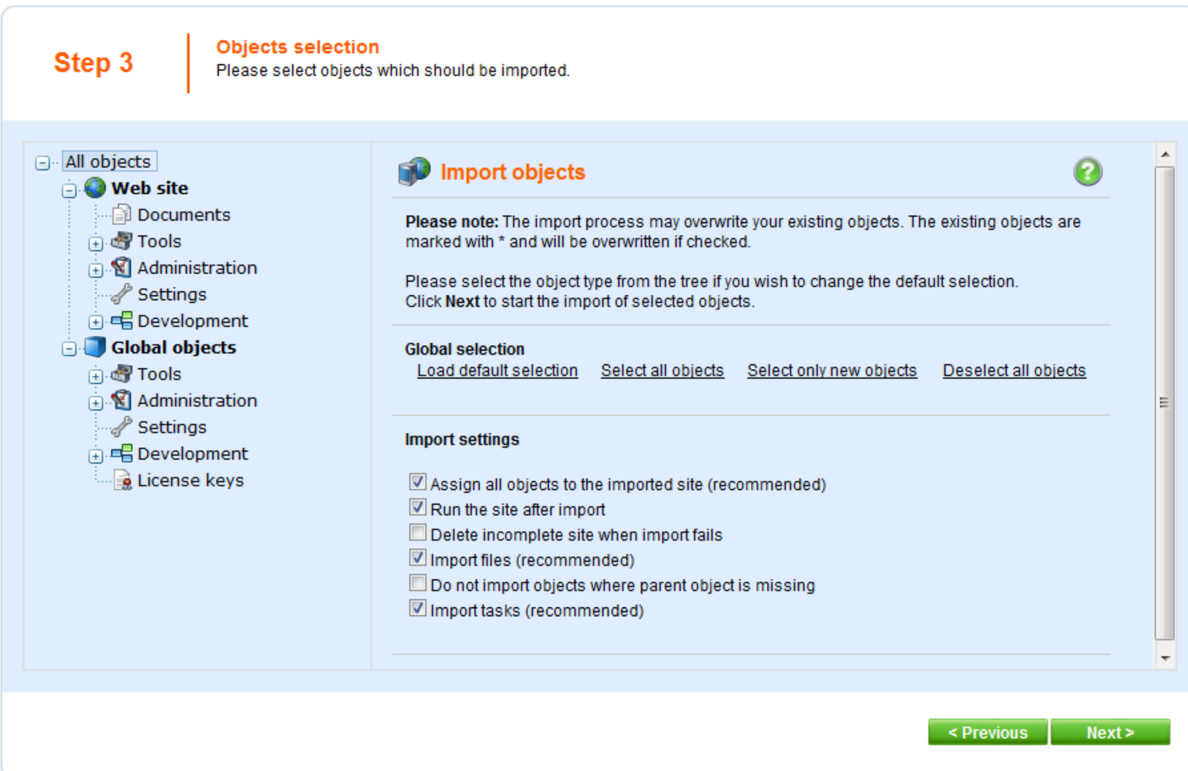
If you select the root of the tree (**Import objects**), a number of global choices will be offered to you. The **Global selection** section allows you to select all, only new or no objects at all. You can also load the default selection, so that all selection changes you've done will be rolled back. The Import settings section contains other basic import settings:

- **Update site definition -** displayed only when importing to an existing site; if enabled, all settings stored as a part of the site object will be updated with those contained in the package; these settings are contained in the **Site\cms_site.xml** file inside the export package
- **Assign all objects to the imported site (recommended)** - if checked, all imported site-related objects will be assigned to the imported site
- **Run the site after import** - if checked, the imported site will be run after the import process finishes
- **Delete incomplete site when import fails** - if checked, incompletely imported site will be deleted when the import process fails
- **Import files (recommended)** - some objects in the database are linked with physical files in the file system; if you check this check-box, such files contained in the package will be imported too
- **Overwrite system queries** - if checked, system queries will be overwritten by those contained in the package
- **Do not import objects where parent object is missing** - if checked, child objects whose parent objects are not found will be skipped and the import process will continue
- **Import tasks (recommend)** - if checked, delete tasks (incremental deployment) included in the

package will be performed

Overwrite system queries

This option is displayed only when importing from version older than the current one; if checked, all queries from the package will be imported and will overwrite the current ones; if the package contains your own **custom queries** that you added to the system, it is necessary to have this option enabled.



The following categories contain extra options to be set:

Documents

- **Import new documents** - if checked, documents will be imported (when importing into an existing site, only new documents will be imported)
- **Import document relationships** - if checked, document relationships will be imported
- **Import document level permissions** - if checked, document security settings made in CMSDesk will be imported
- **Import blog comments** - if checked, blog comments will be imported

Please note

When importing into an existing site, only new documents can be imported. Modified documents that are already present on the target server will not be overwritten.

BizForms

- **Import BizForms data -** if checked, BizForms data included in the package will be imported together with the BizForms

Forums

- **Import forum posts -** if checked, forum posts included in the package will be imported together with the forums

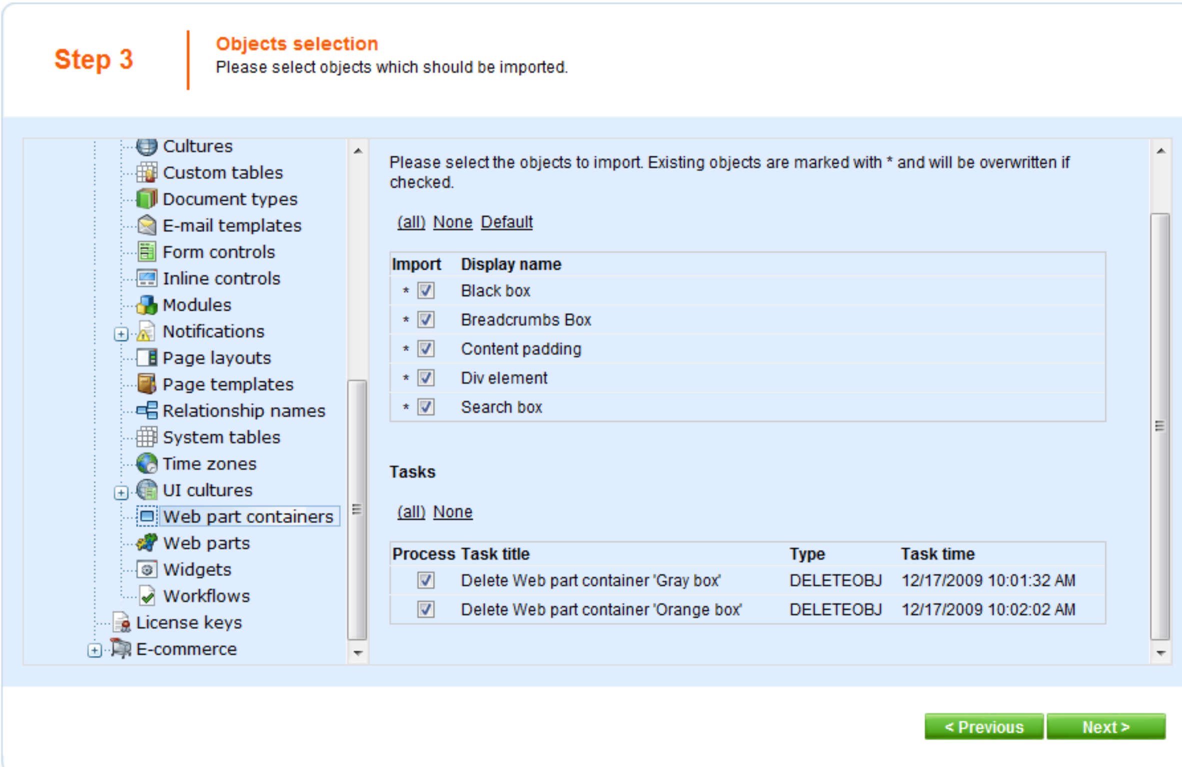
Message boards

- **Import board messages -** if checked, board messages included in the package will be imported together with the message boards

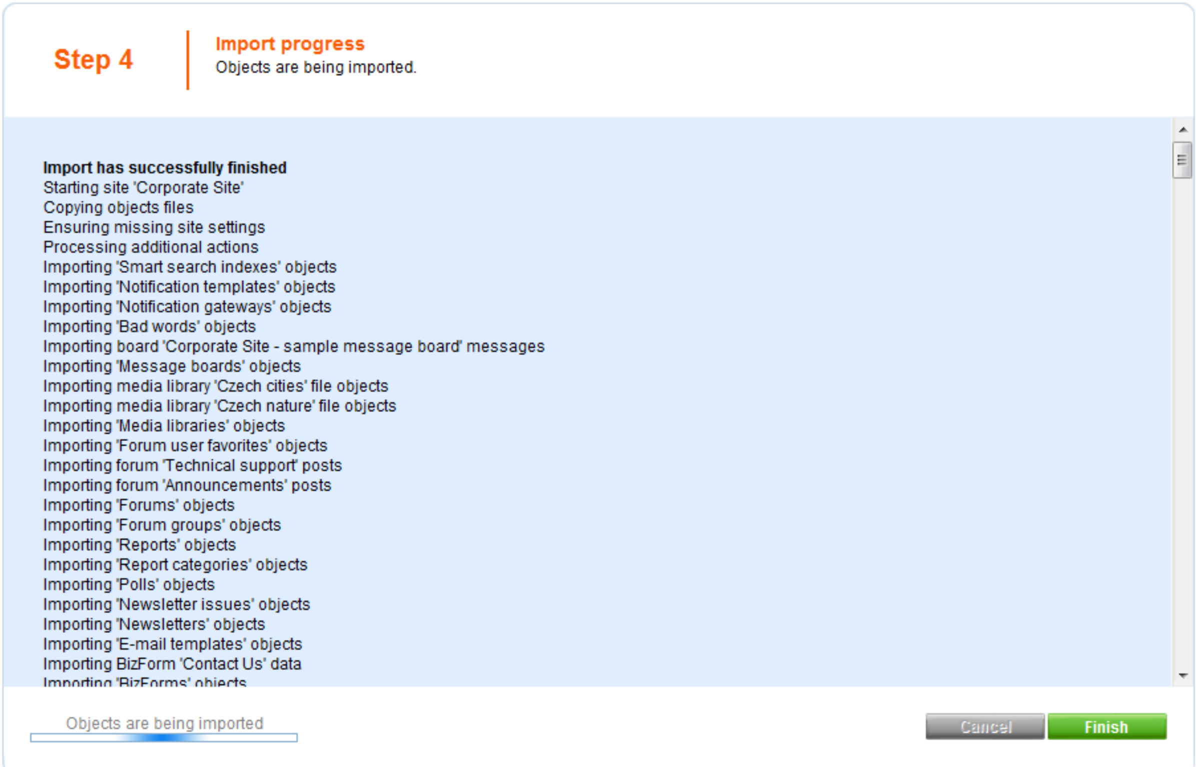
Media libraries

- **Import media files -** if checked, media files (stored in database) included in the package will be imported together with the media libraries
- **Import physical files -** if checked, physical files (stored in the file system) included in the package will be imported together with the media libraries

If you have the "**Log export tasks**" option enabled in **Site Manager -> Settings -> Content Staging**, a list of tasks may also be displayed under the objects list. This happens when some global objects were deleted (just as the two Web part containers in the screenshot below). If you leave the check-boxes checked, these objects will be deleted on the target server.



An import log will be displayed, showing the progress of importing (you can abort importing by clicking the **Cancel** button any time). Click the **Finish** button. You will be redirected back to **Site manager** -> **Sites**, where the newly imported site will be running.

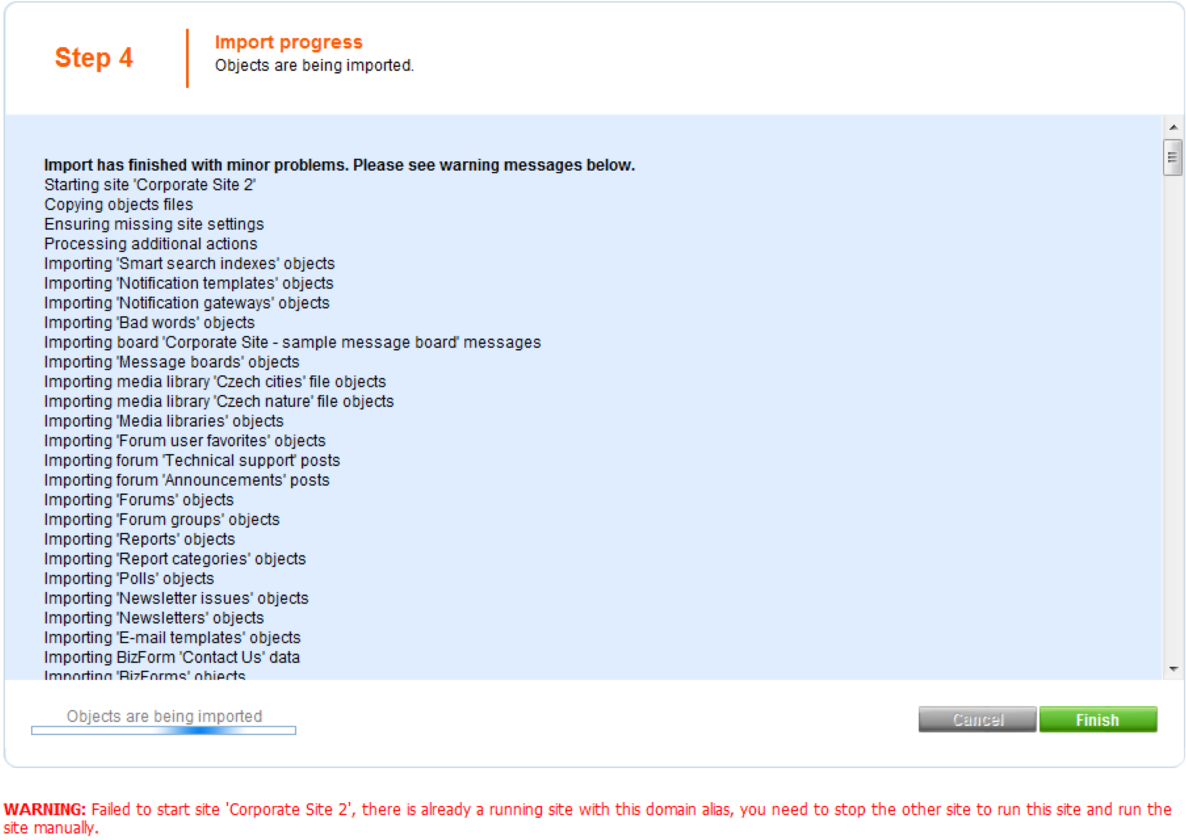


Please note

Packages from different versions of Kentico CMS have different structure. When importing packages from an older version of Kentico CMS to a newer one, structure of the package is always converted to the newer format automatically.
Please pay special attention when importing **Form user control**, **Inline control** and **Web part** objects from older packages. If possible, avoid overwriting your current objects of these types with objects from older packages, as it may cause incompatibility problems.

Conflicts of running sites

If the imported site uses the same domain name or alias as one of the websites that already run on your server, you may get an error message at the end of the import:



In such case, you need to go to the **Sites** section, change the domain name or domain alias and start the new website manually using the **Start site** button.

Application restart

At the end of the import process, you may get the following error message:

"Application has been restarted and the logging of the import process has been terminated. Please see context help in this section for more details and how to solve this issue."

If so, you will have to finish the import process manually:

1. Open the imported package and extract the following folders located in *<package>\Data\Files* (if they are present):
   - cms_webpart
   - cms_cssstylesheet
   - cms_documenttype
   - cms_formusercontrol
   - cms_inlinecontrol
   - cms_pagetemplate
   - Global

Also extract the *<package>\Data\Site* folder.

2. These folders contain subfolders named identically to the folders inside your web

project root. Remove the '.export' extensions from the included files and copy them manually to the appropriate location inside your web project folder with respect to the original folder structure.
3. Under the Site folder, the *##SITENAME##* folder may be found in the folder structure. If so, please rename this folder to the name of the imported website and copy the content of the Site folder to the root of the web.

5.5.6 Folder structure and import/export

Kentico CMS uses a single folder structure, even if you manage multiple websites in a single installation. The following list describes the main folders and how they are affected during the import and export:

- **App_Browsers**
- **App_Code**
- **-** CMSModules**<module_name>** (folders of custom modules)
- - **Global** (exports with any site, needs to be created manually; the folder is exported if the 'Export global folders' option is checked in Step 2 of the export process)
  - **<site code name>** (exports with given site, needs to be created manually;the folder is exported if the 'Export site folders' option is checked in Step 2 of the export process)
- **App_Data**
- **-** CMSModules**<module_name>** (folders of custom modules)
- **App_Themes**
  - **<stylesheet name>** (all folders related to stylesheets assigned to or used on the website)
- **App_WebReferences**
- **aspnet_client**
- **bin**
- **ClientBin**
- **CMSAdminControls**
- **CMSControlsExamples**
- **CMSDesk**
- **CMSFormControls** (all form controls selected in Step 2 of the export process are exported with any site)
- **CMSGlobalFiles** (exports with any website, needs to be created manually; folder is exported if the 'Export global folders' option is checked in Step 2 of the export process)
- **CMSImportFiles**
- **CMSInlineControls** (all form controls selected in Step 2 of the export process are exported with any site)
- **CMSInstall**
- **CMSMasterPages**
- **CMSMessages**
- **CMSModules**
- - Forums\Controls\Layouts**Custom** (forum custom layouts; exported with any website, needs to be created manually;folder is exported only if the 'Copy forum custom layouts folder' option is checked in Step 2 of the export process)
- - **<module name>** (folders of custom modules)
- **CMSPages**
- **CMSResources**
- **CMSScripts**
- **CMSSiteManager**
- **CMSSiteUtils**
- **CMSTemplates** (all files for selected ASPX page templates are exported with any site, page

templates in other folders are exported as well if they are assigned/used in the given site;whole folder is exported if the 'Export ASPX templates folder' option is checked in Step 2 of the export process)

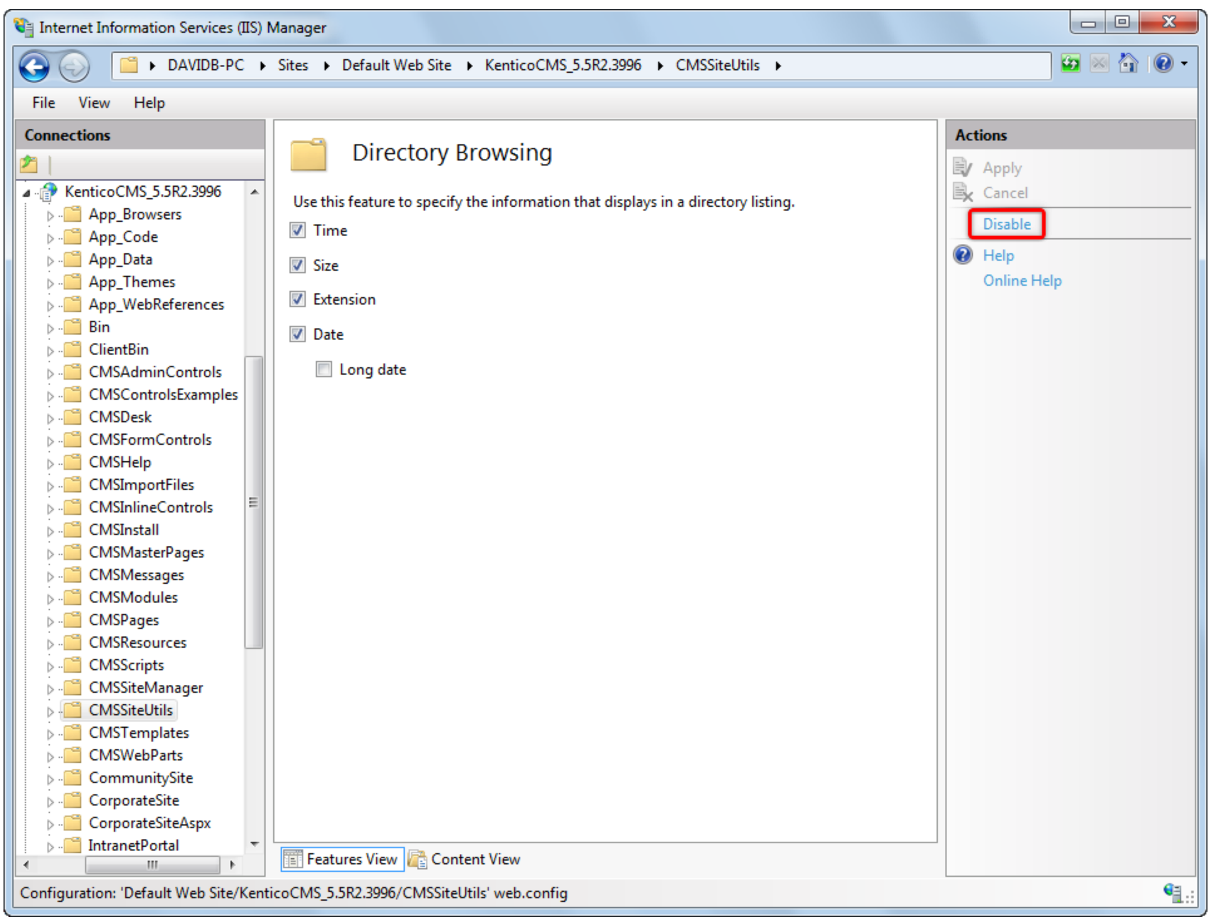
- **CMSWebParts** (all web parts selected in Step 2 of the export process are exported with any site, if the web part uses some additional files, they must be placed in the <webpartCodeName>_files folder)
- **<site code name>** (exports with given site, needs to be created manually or may be created automatically when storing files on the disk; the folder is exported if the 'Export site folders' option is checked in Step 2 of the export process)
  - Files (default folder for storing files if the system is configured for saving files on the disk)

Here's the explanation of colors:

- **red** - system folder, do not make changes or place your files here unless you want to modify the administration interface
- **blue** - folders for custom files, part of the export package
- **green** - folders for custom files, part of the export package, may need to be created manually
- **black** - service folders (import files, import/export)

Export/Import package security

It is highly recommended to disable **Directory Browsing** in IIS for websites on live servers, at least for the **CMSSiteUtils** directory. If enabled, sensitive data from site export/import packages could be accessed directly from the browser, such as for example user credentials.



5.5.7 Excluding files and folders from export

Files and folders that are exported into the **Files** folder of the export package can be filtered to be excluded from export. This is achieved by adding the following keys into the *AppSettings* section of your site's *web.config* file:

- **Excluding folders from export**: `<add key="CMSExportExcludedFolders" value="test*;cms*" />`
- **Excluding files from export**: `<add key="CMSExportExcludedFiles" value="test*;cms*" />`

Values of the keys define names of files/folders which will be excluded from export, i.e. will not be exported. Multiple values can be entered separated by semicolons.

You can also use the standard file system mask using the * wildcard, which substitutes any number of characters in the name. No other file system mask wildcards are supported.

.scc files and **.svn** folders are excluded by default, even if the keys are not entered in the *web.config* file.

5.5.8 Import/export troubleshooting

5.5.8.1 Configuration on W2008/IIS7

KenticoCMS fully supports **Microsoft Windows Server 2008** and **Internet Information Server 7**. Should you experience difficulties with import/export on IIS7, please go through the following steps:

1. Open **IIS manager**.

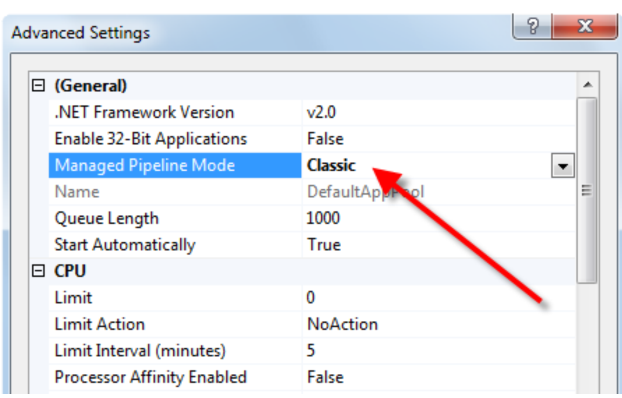
2. Open **Application pools** section.

3. Select application pool of your site and select **Advanced settings...** .

4. For property **Managed Pipeline Mode** select **Classic** instead of **Integrated**

or

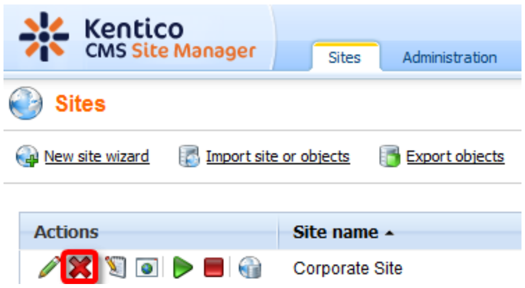
You can select **Classic .NET AppPool** for your website.



5.6 Deleting sites

You can delete sites in the system in **Site Manager -> Sites**.

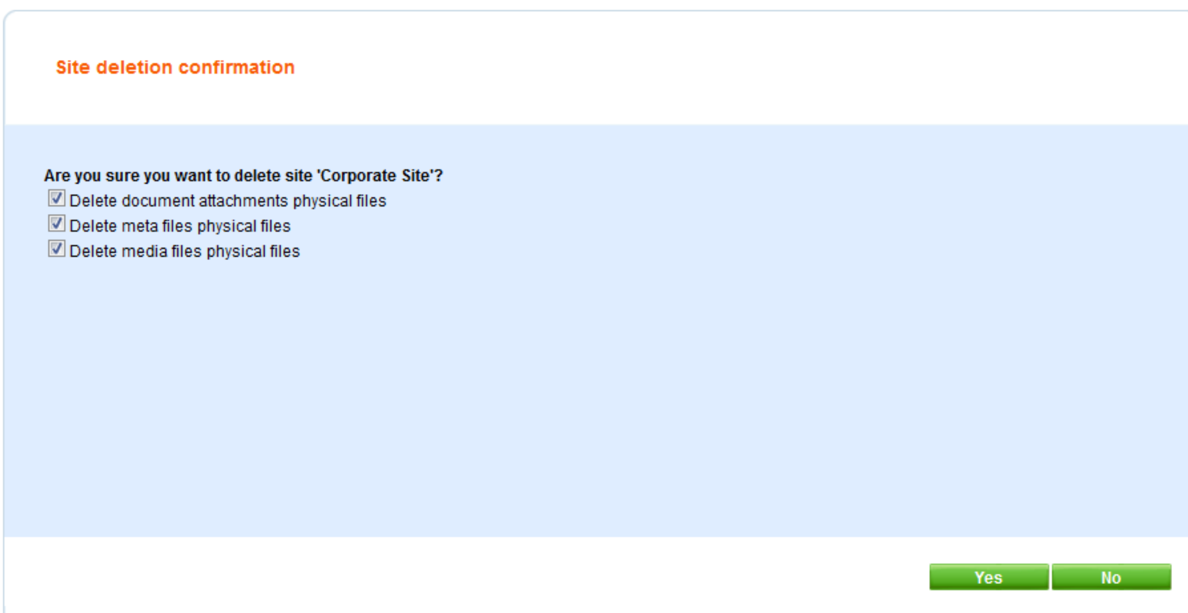
1. Clicking the **Delete** (✖) icon of the site that you want to delete.



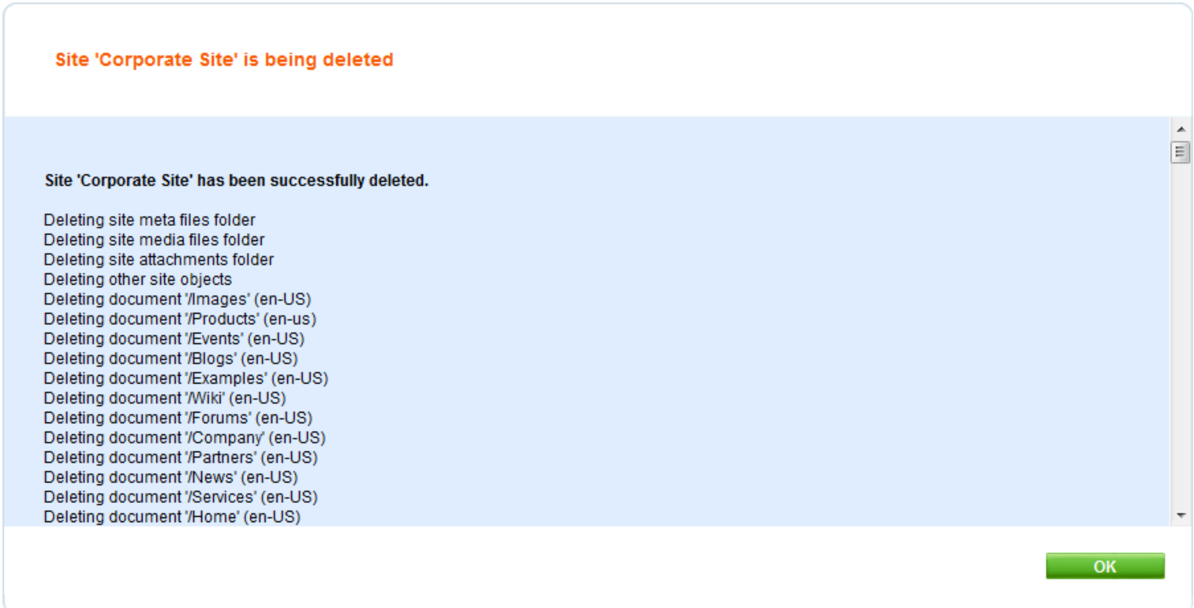
2. In the **Site deletion confirmation** dialog, you can select the following options:

- **Delete document attachments physical files** - if checked, document attachment files stored in the file system will be deleted; these files are stored in the <*web project*>\<*site name*>*files* folder
- **Delete meta files physical files** - if checked, meta files stored in the file system will be deleted; these files are stored in the <*web project*>\<*site name*>*metafiles* folder
- **Delete media files physical files** - if checked, physical files stored in media libraries will be deleted; these files are stored in the <*web project*>\<*site name*>*media* folder

Make the selection and click **Yes** to continue deleting the site.



3. A log will be displayed, showing you the progress of site deletion. When the process finishes, click **OK**. You will be redirected back to **Site Manager -> Sites**, where the deleted site will not be listed.

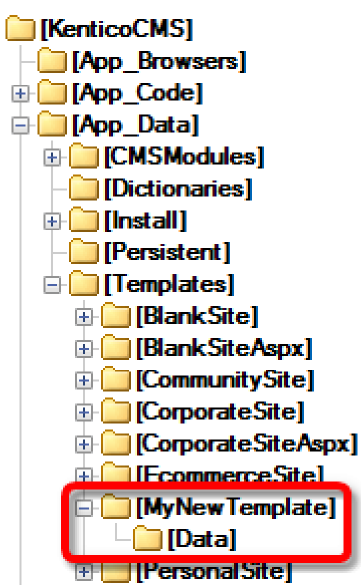


5.7 Creating web templates

In case that you want to use your current site created with Kentico CMS as a web template, so that you can use it a a starting point for developing new sites, you have to take the following steps:

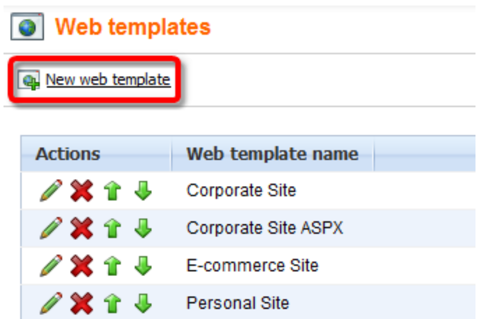
1. Export your site. For a step-by-step tutorial on how to export a site, please refer to the Exporting a site chapter of this guide.

2. Go to **<web project>\App_Data\Templates**. As you can see, this is the folder where all the default templates, such as Community or Corporate site, are stored. Create a new folder with the name of your new page template. Then create one sub-folder under the newly created folder and give it the name **Data** .



3. Extract the content of your export package into the **Data** folder.

4. Go to **Site Manager -> Development -> Web templates** and click the **New web template** link.



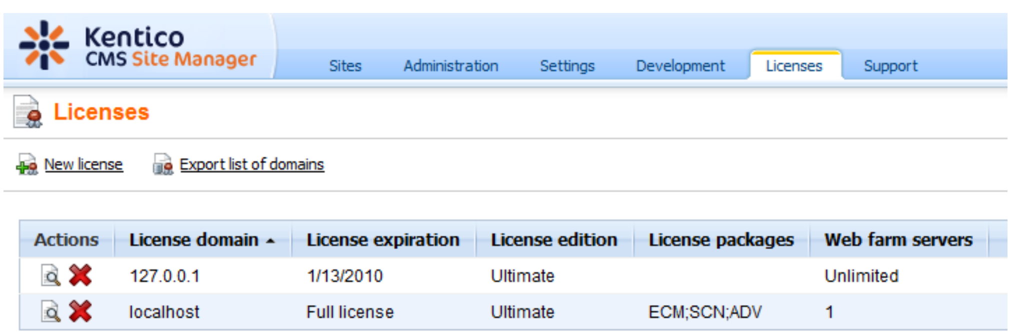
5. Enter the following details:

- **Web template display name** - name of the web template displayed in the administration interface
- **Web template code name** - name of the web template used in code
- **Web template folder name** - path to the folder where you have extracted the content of the export package; *~\App_Data\Templates\<your folder>*
- **Web template description** - text describing your new web template
- **License editions** - editions of Kentico CMS in that this web template will be available; check all for full availability

and click **OK**. Your new web template should now be present in the list.

5.8 License management

Kentico CMS requires an appropriate license key for every domain you use. The licenses can be managed in **Site Manager -> Licenses**:



The list displays the information about licensed domain, expiration and edition. When you get full or trial key for a particular domain, you need to click **New license** ( ) and enter the full text of the key into the **License key** field.

You can also use the **Export list of domains** ( ) link to export your domains. The export package will be saved in *~\CMSSiteUtils\Export* and can be used to import the licenses on another instance of Kentico CMS.

How licensing works

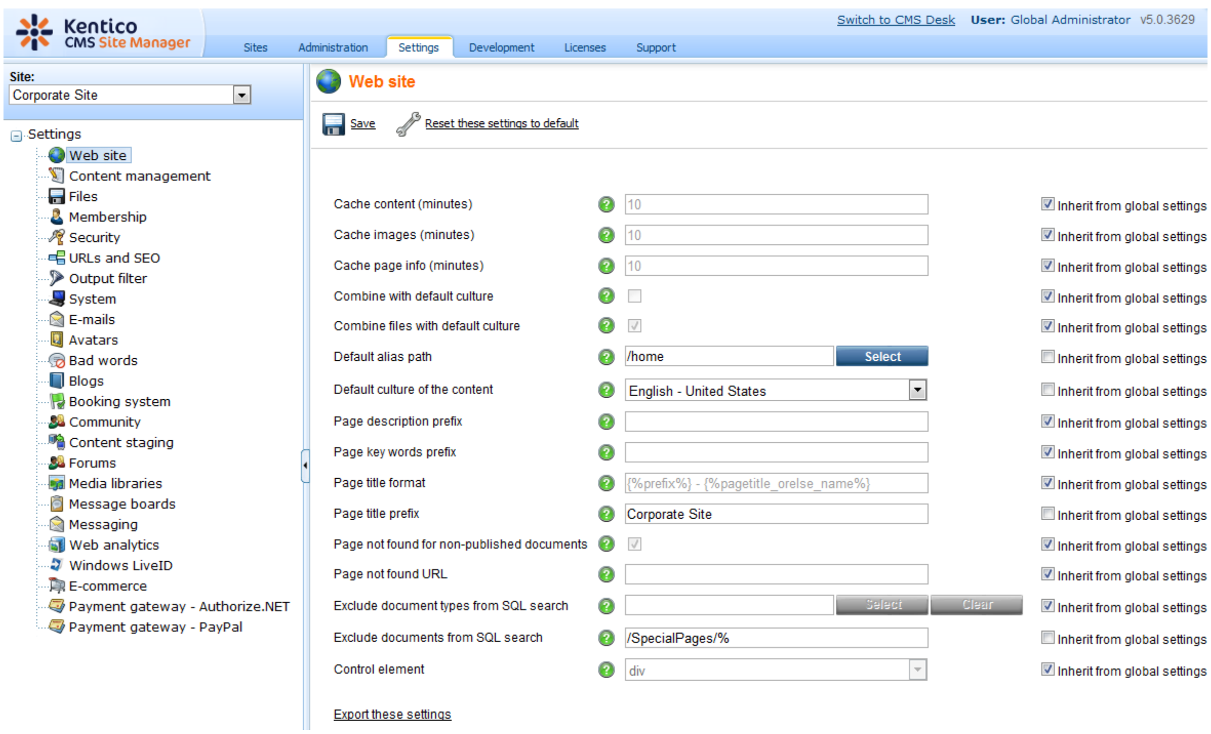
If you're running website on domain example.com, you need a single license key that will also work for:

- http://example.com
- https://example.com
- http://www.example.com
- https://www.example.com
- http://localhost
- http://127.0.0.1

If you use a domain alias (different domain name that points to the same website), such as *example1.com* or *example.net*, you need extra license keys for these domain aliases. Please ask [Kentico support](#) for generating the additional keys (they are free of charge if you already own a license for the main domain).

5.9 Managing site settings

Most of the site settings can be configured in **Site Manager -> Settings** section.



There are two basic types of settings:

- **Global** – such settings apply to all sites.
- **Site-specific** – such settings apply to the particular site and they override the global settings values.

If you want to inherit value from the global settings, you need to check the "inherit from global settings" button and click Save.

Tip: If you mouse-over the name of the settings key, you will see the description of the key.

5.10 Configuring multiple web sites

Kentico CMS allows you to run multiple websites from a single installation (code base) and database. All websites run as a single website in IIS. The following tutorial explains how to set up multiple websites on Windows XP and Windows Server 2003.

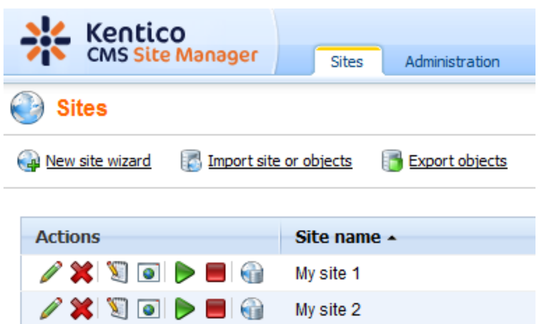
We will configure two websites:

- **mysite.com**
- **mysite2.com**

Configuring multiple sites in Kentico CMS (common for all operating systems)

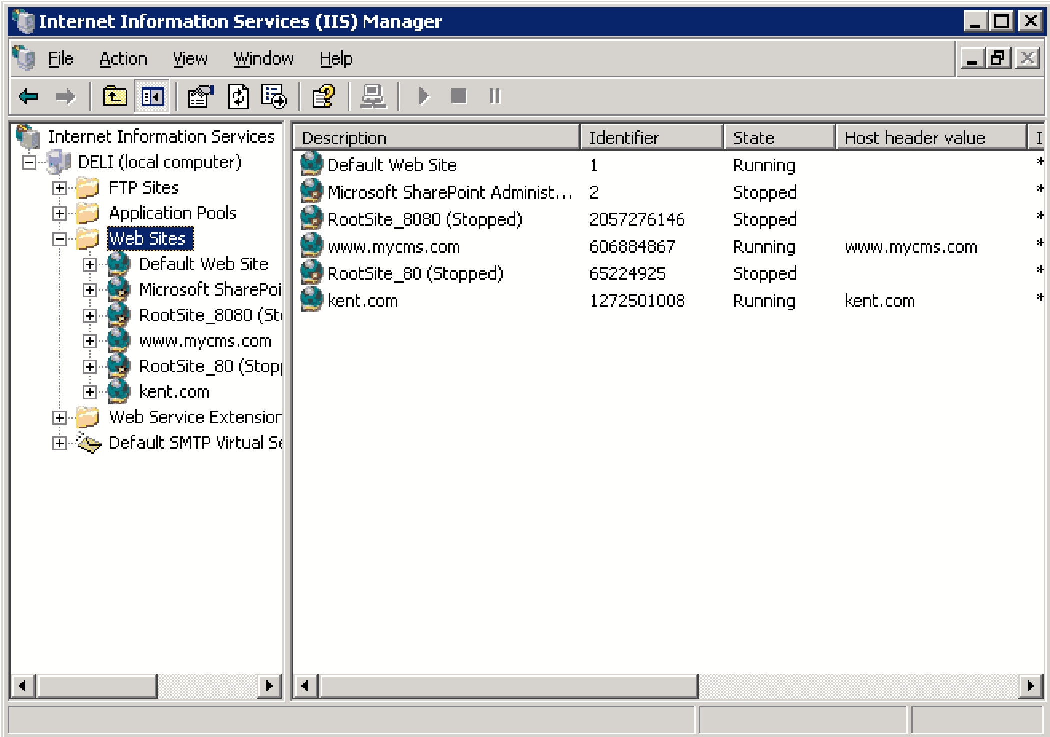
This part is common for all operating systems.

1. Create two websites in **Site Manager -> Sites** - you can either import your existing websites or you can create new websites using the New site wizard.
2. Edit the properties of each website in **Site Manager -> Sites** and set the **Site domain name** value of the each website to the appropriate domain (without www prefix and without http:// protocol).
3. Make sure both sites are running. You can check this in the Site list, in the Status column.
4. Make sure you have valid license keys for both domains in **Site Manager -> Licenses**.

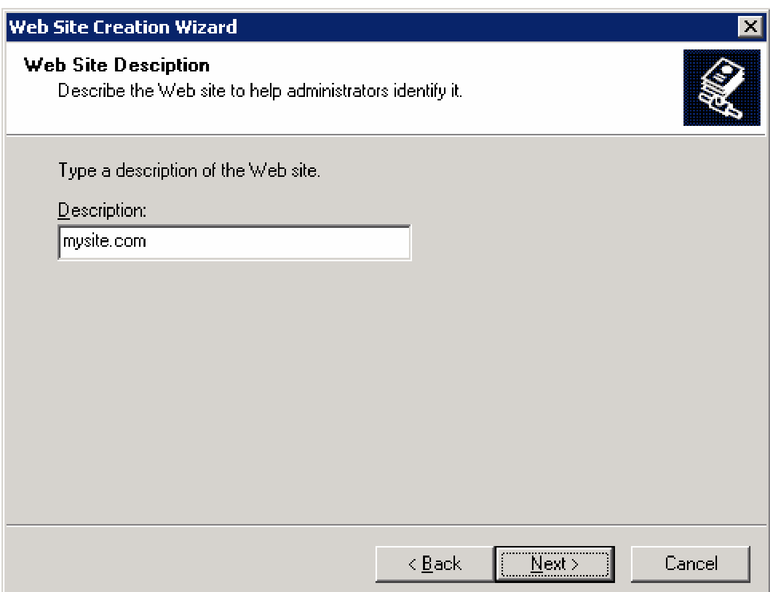


Configuring multiple sites on Windows Server 2003

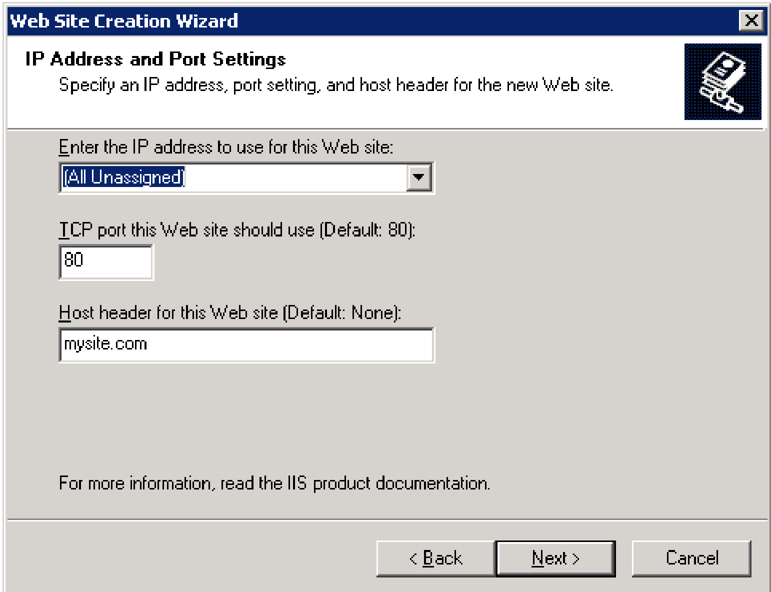
Open **Start -> Administrative tools -> Microsoft Internet Information Services (IIS) Manager**, go to websites section and if the website does not exist yet, create it.



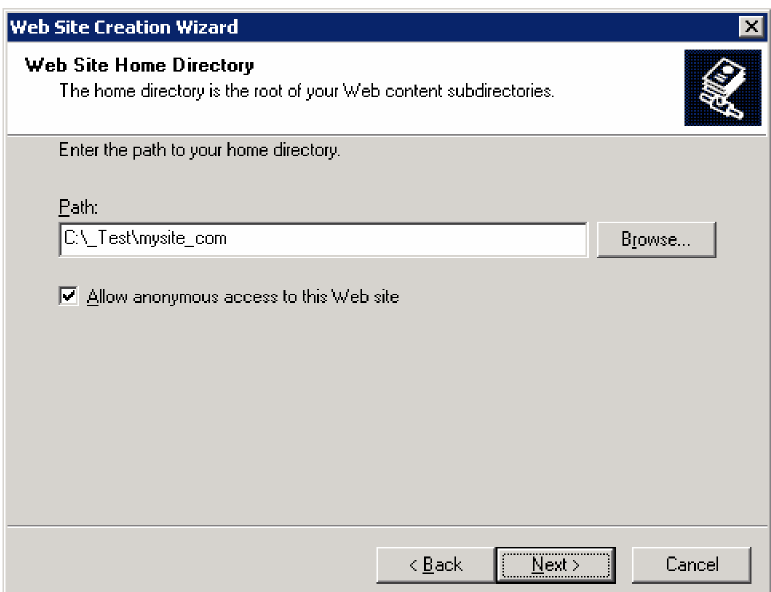
Right-click **websites** and choose **New -> website...** The website Creation Wizard starts. Enter the site descriptive name, such as **mysite.com**:



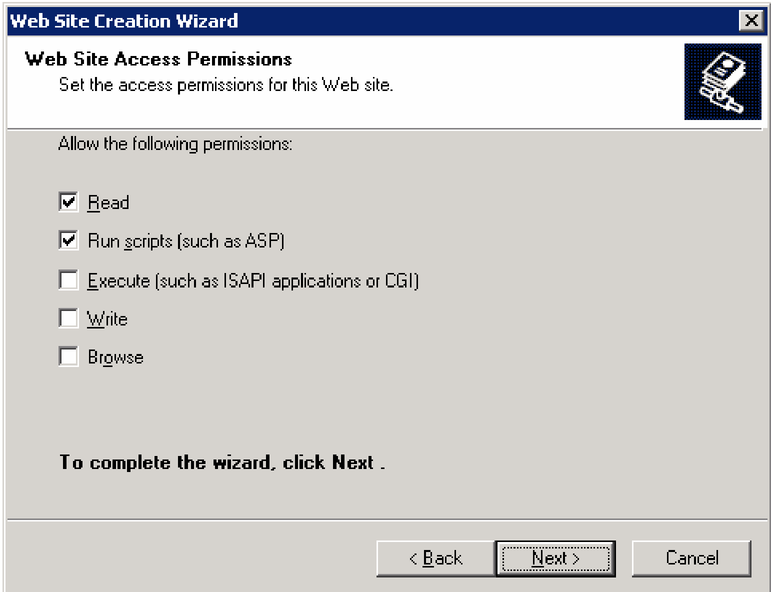
Click **Next**. Enter the **Host header for this website** as **mysite.com**:



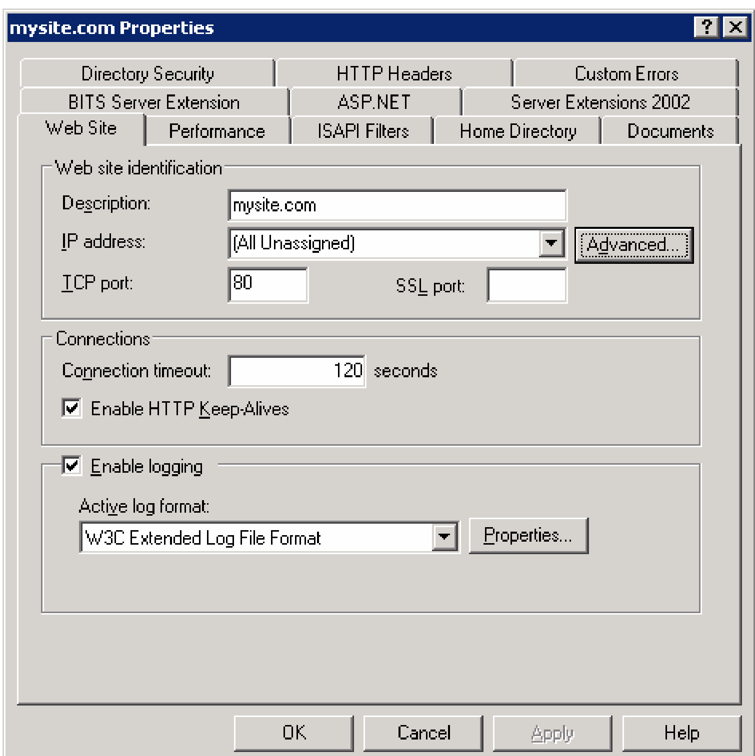
Click **Next**. Select the disk path where the root of your website is placed. This must be the folder where web.config file is placed.



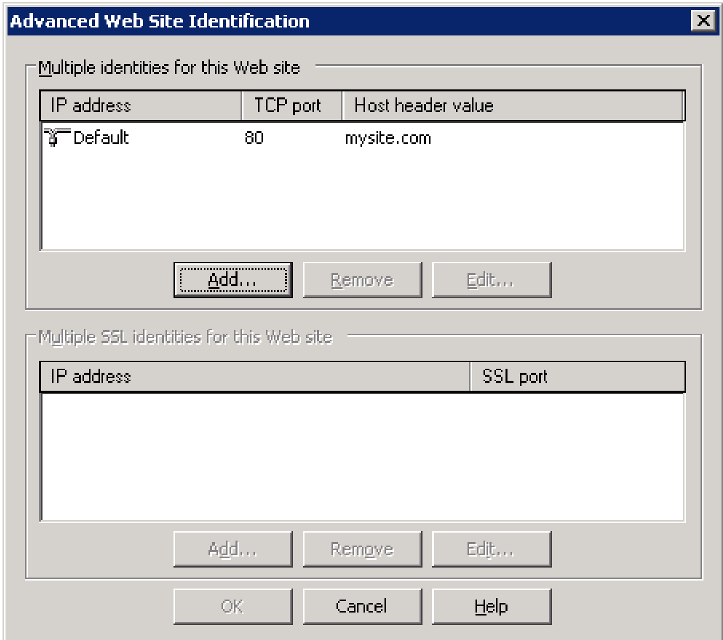
Click **Next**. In the next step, enable the **Read** and **Run scripts** boxes.



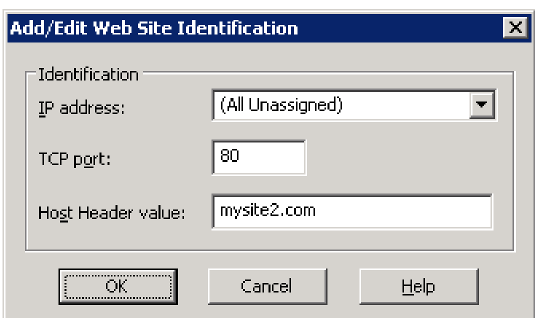
Click **Next**. The wizard is finished. Click **Finish**. Now we need to add the other domain name to the list of URLs hosted by this website. Right-click the newly created website and display website properties. On the **website** tab, click **Advanced...**



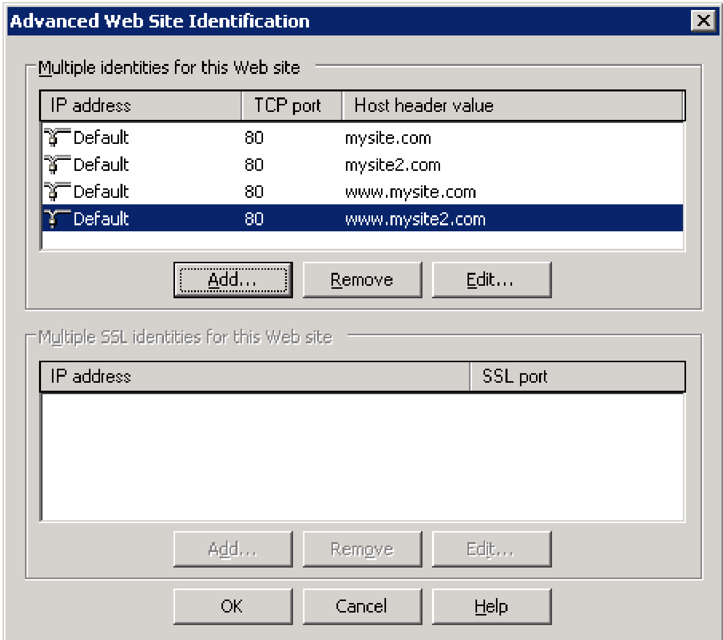
Now you need to add another host header value for your second domain:



Click the top **Add...** button and enter the appropriate values. The standard HTTP port is 80:

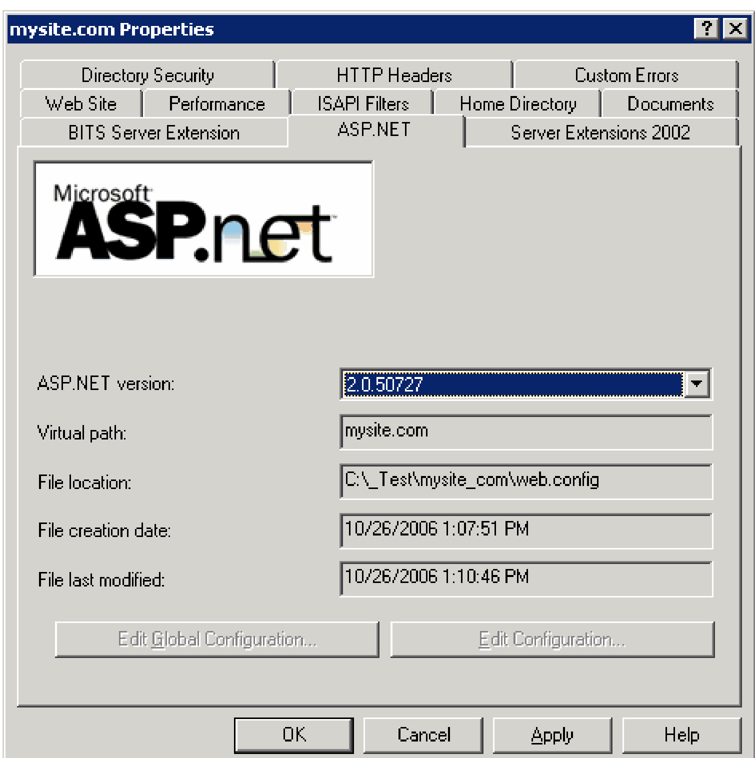


You need to repeat this also for www.* alternatives of your website:



Click **OK** on all dialogs.

Please note: You may also need to configure the website for ASP.NET 2.0 in website Properties, on the **ASP.NET** tab. IIS 6 may display the **ASP.NET version as 2.0.50727** even if you have **ASP.NET 3.5** or higher installed and registered:



Your new website is now configured to host all incoming requests for domains mysite.com and mysite2.

com (or other domains depending on your particular situation). You may need to ask your network administrator to redirect the domain in the DNS records to your website.

If you do not own the domain, you can test it by modifying the C:\WINDOWS\system32\drivers\etc\hosts file in notepad and adding the following lines to the end of the file:

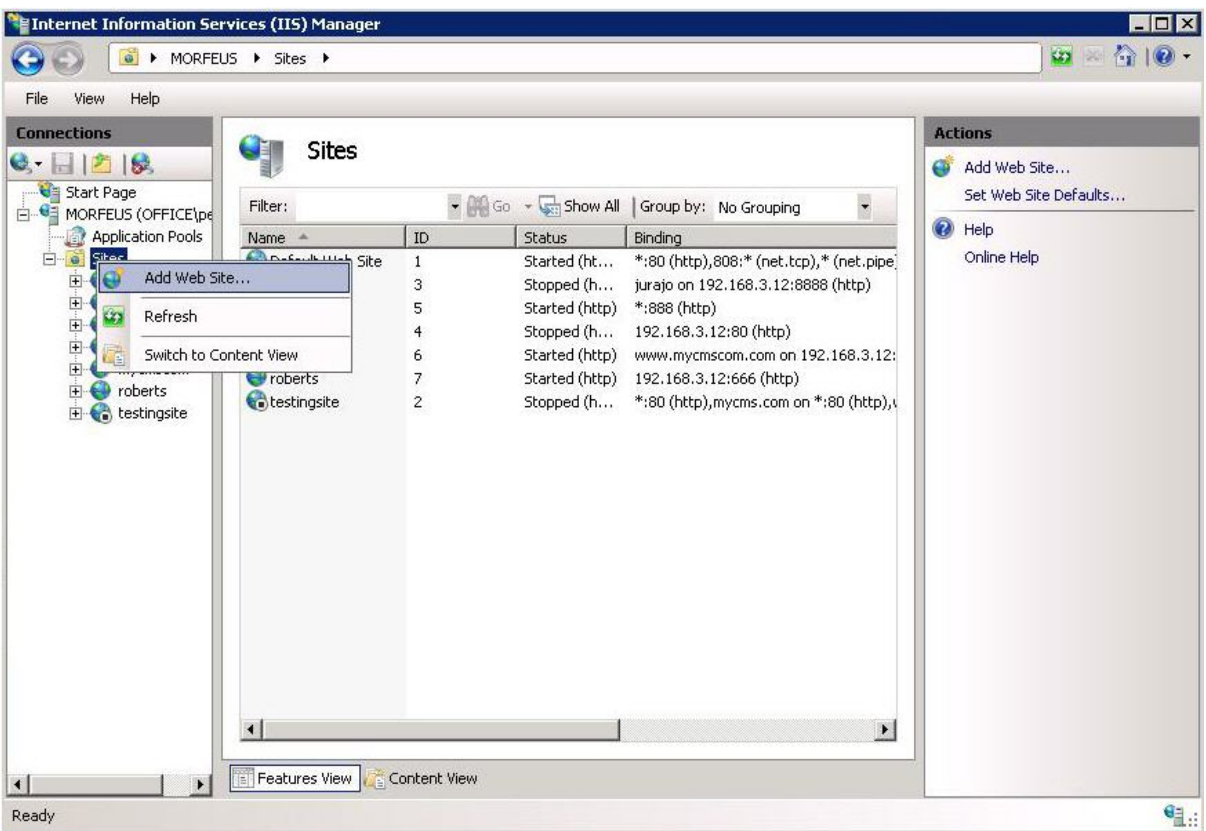
```
127.0.0.1       mysite.com
127.0.0.1       www.mysite.com
127.0.0.1       mysite2.com
127.0.0.1       www.mysite2.com
```

Save the file. Please note: these are client settings, which means they will work only if you use web browser on your server.

Now, when you go to *http://www.mysite.com* and to *http://www.mysite2.com* (or your own domain names), you should see two different websites.

Configuring multiple sites on Windows Vista / Server 2008

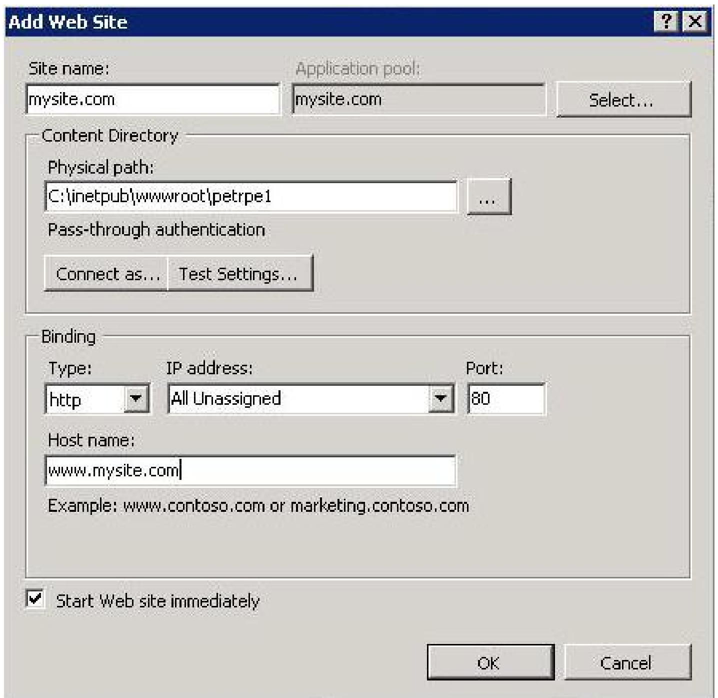
Go to **Start -> Control Panel -> Administrative tools -> Microsoft Internet Information Services (IIS) Manager.** In the tree view, right click **Sites** and choose **Add website...**.



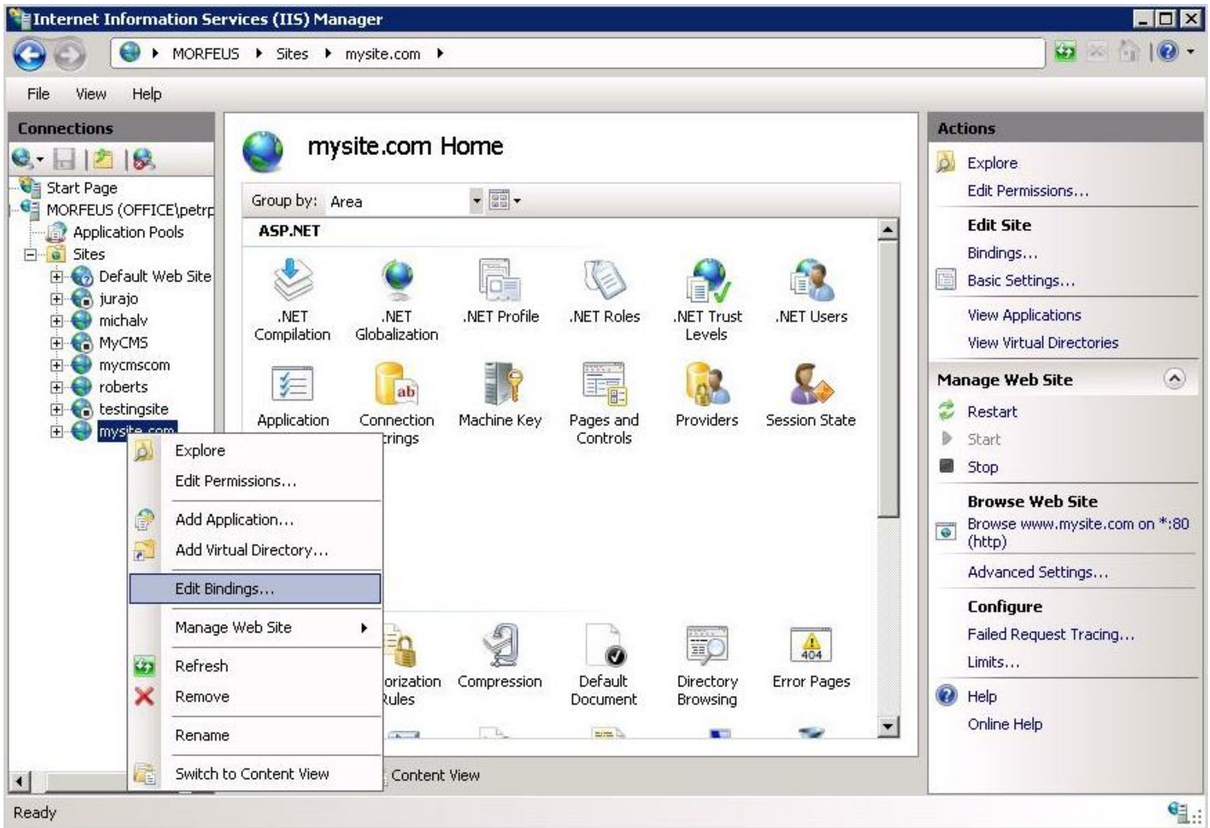
Enter the following details:

- **Site name**: mysite.com
- **Physical path**: disk path to the location where your website is placed; this must be the location where the *web.config* file is stored
- **Host name**: www.mysite.com

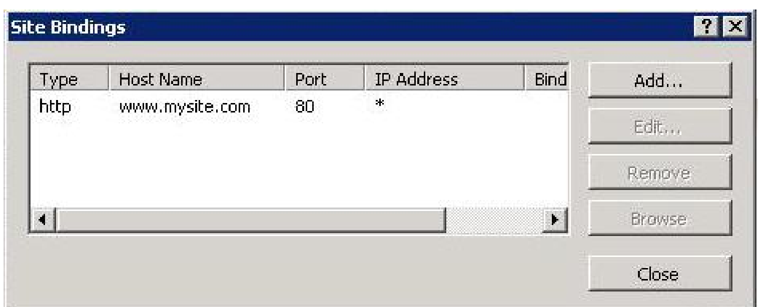
Click **OK**.



The site should now appear in the tree, under the **Sites** node. Right click the site and choose **Edit bindings** (or **Bindings** on Vista).

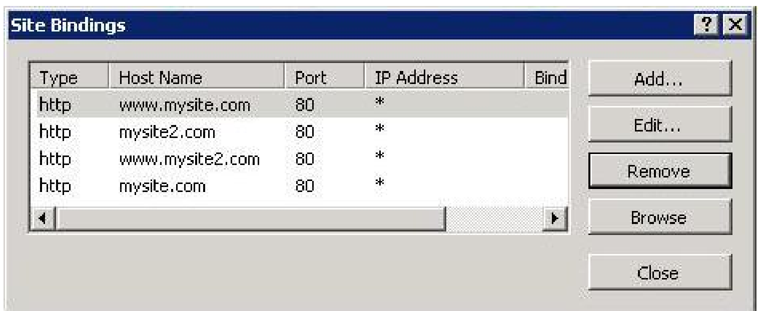


The site bindings dialog appears. Click **Add...**.



Enter the domain name of your second website (*www.mysite2.com* in this case) into the **Host name** field and click **OK**. Repeat this for both of the sites without the '*www.*' prefix.

The result should look like the following screenshot.



Your new website is now configured to host all incoming requests for domains mysite.com and mysite2.com (or other domains depending on your particular situation). You may need to ask your network administrator to redirect the domain in the DNS records to your website.

If you do not own the domain, you can test it by modifying the C:\WINDOWS\system32\drivers\etc\hosts file in notepad and adding the following lines to the end of the file:

```
127.0.0.1       mysite.com
127.0.0.1       www.mysite.com
127.0.0.1       mysite2.com
127.0.0.1       www.mysite2.com
```

Save the file.

Please note: these are client settings, which means they will work only if you use web browser on your server.

Now, when you go to http://www.mysite.com and to http://www.mysite2.com (or your own domain names), you should see two different websites.

Configuring mulitple sites on Windows XP

On Windows XP, the support of multiple sites and domains in IIS is limited, so we will use a single IIS website and define "virtual" domains in our hosts file:

Open the c:\WINDOWS\system32\drivers\etc\hosts file in notepad and add the following lines to the end of the file:

```
127.0.0.1        mysite.com
127.0.0.1        www.mysite.com
127.0.0.1        mysite2.com
127.0.0.1        www.mysite2.com
```

Save the file. Please note: these are client settings, which means they will work only if you use web browser on your local computer.

Go to http://www.mysite.com or http://www.mysite.com/kenticocms (where kenticocms is the name of the virtual directory with Kentico CMS website) and to http://www.mysite2.com. You should see two different websites.

Multiple websites on a single domain (in subfolders)

If you cannot use (for some reason) multiple domain names, you can configure Kentico CMS so that it differentiates websites by subfolder (virtual directory). Read chapter Multiple websites on a single domain (in subfolders) for more details.

Using multiple websites on Windows XP

Windows XP allows you to run only one IIS website at a time. If you need to develop multiple websites (multiple Kentico CMS instances) in the root folder, you may need to create additional websites and switch between them using the IISAdmin utility that can be downloaded at http://jetstat.com/iisadmin/download.asp.

5.11 Multiple web sites on a single domain (in subfolders)

In some cases, you may want to run multiple websites in separate subfolders, without getting a new domain.

In this case, you can configure the domain names (or domain aliases) of the websites like in the following example:

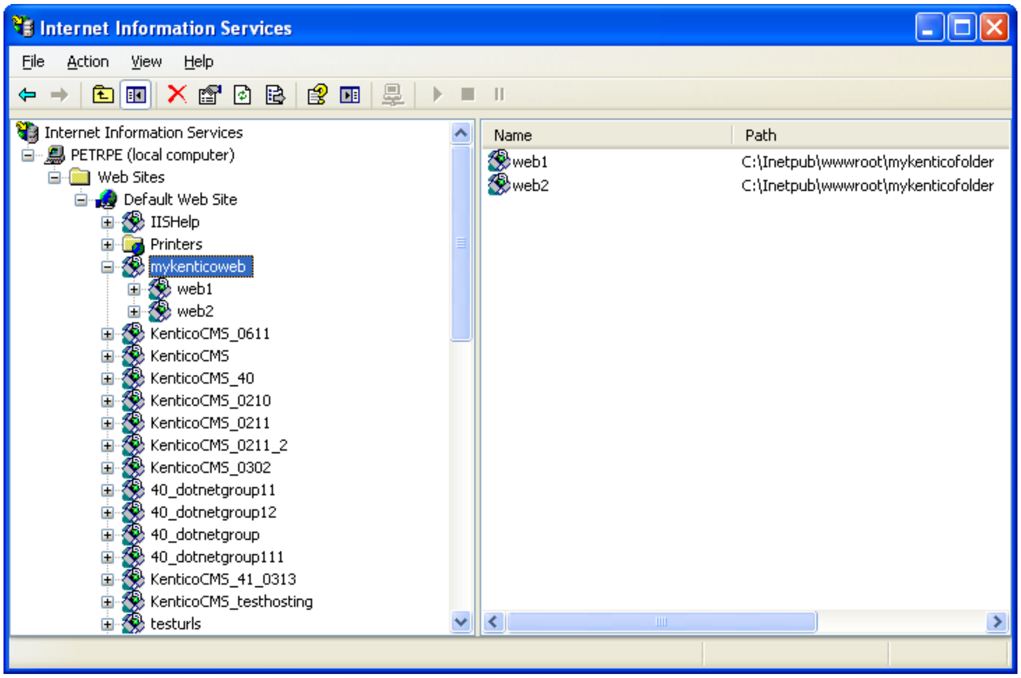
Example

1. Install Kentico CMS to the following folder: **C:\inetpub\wwwroot\mykenticofolder**. Make sure the name of the folder is different from the name in the URL that you will later enter in IIS. In the same step of the installer, choose **This is an installation to the root (do not create virtual directory)**. When the setup finishes, the link to your new website will not work as the virtual directory is not created in IIS.

2. Open IIS console (Control Panels -> Administrative Tools -> Internet Information Services) and create a new virtual directory named **mykenticoweb**. Assign it to a **non-website-root folder** on your disk. Ideally, create an empty folder on the disk for this purpose (e.g. c:\empty).

3. Create two new virtual sub-directories under **mykenticoweb** called **web1** and **web2**. Both of them

need to be assigned to the installation folder on the disk, which is **C:\inetpub\wwwroot\mykenticofolder** in this example.



4. Now open your browser and type in either **http://localhost/mykenticoweb/web1** or **http://localhost/mykenticoweb/web2**. Kentico CMS Database setup appears. Continue through the setup as usual and install the first site.

5. When the setup finishes, go to **Site Manager -> Sites** and install the second site.

6. Configure website 1 domain name as: **localhost/mykenticoweb/web1**

7. Configure website 2 domain name as: **localhost/mykenticoweb/web2**

Now when you go to **http://localhost/mykenticoweb/web1**, you will see website 1. If you go to **http://localhost/mykenticoweb/web2**, you will see website 2.

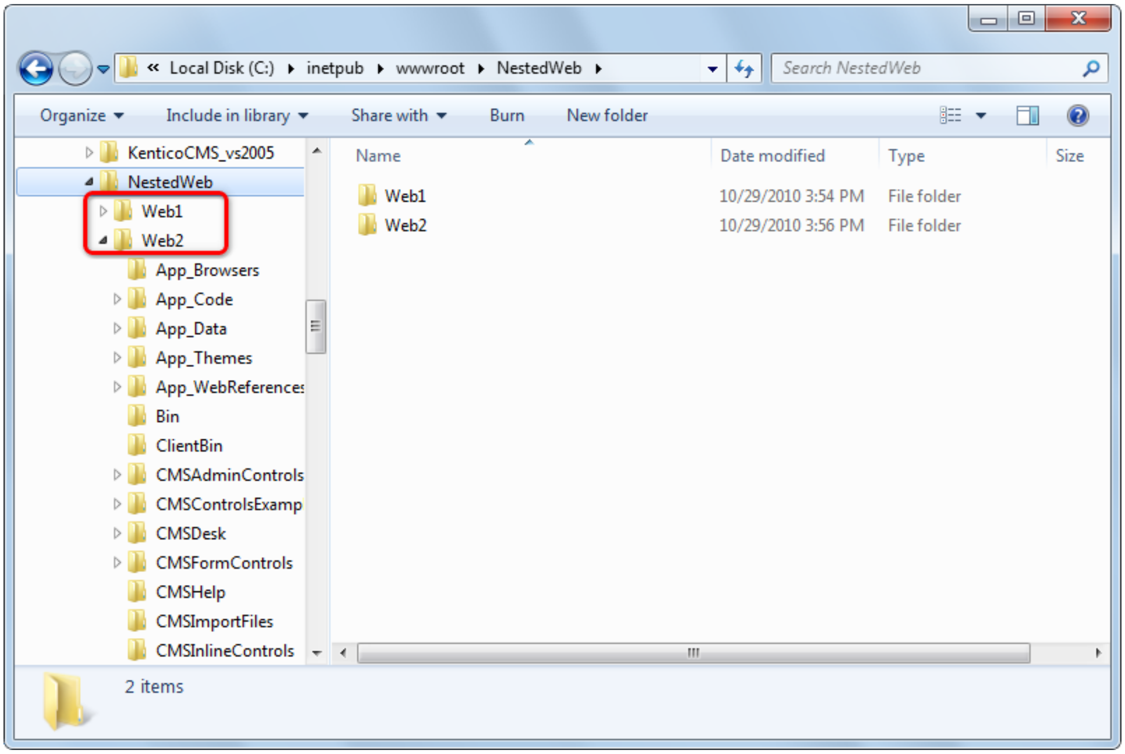
5.12 Configuring nested web sites

In the following example, you will learn how to set up two nested websites. It should be possible to achieve any level of nesting in case that the following steps are complied with.

1. Install two projects into two independent, not overlapping folders. Choose folder names that won't be the same as the names in the URL to prevent later colision when seting up the IIS. The installation folders may be for example like this:

- **Inetpub/wwwroot/NestedWeb/Web1** (first website from Web installer)
- **Inetpub/wwwroot/NestedWeb/Web2** (second website from Web installer)

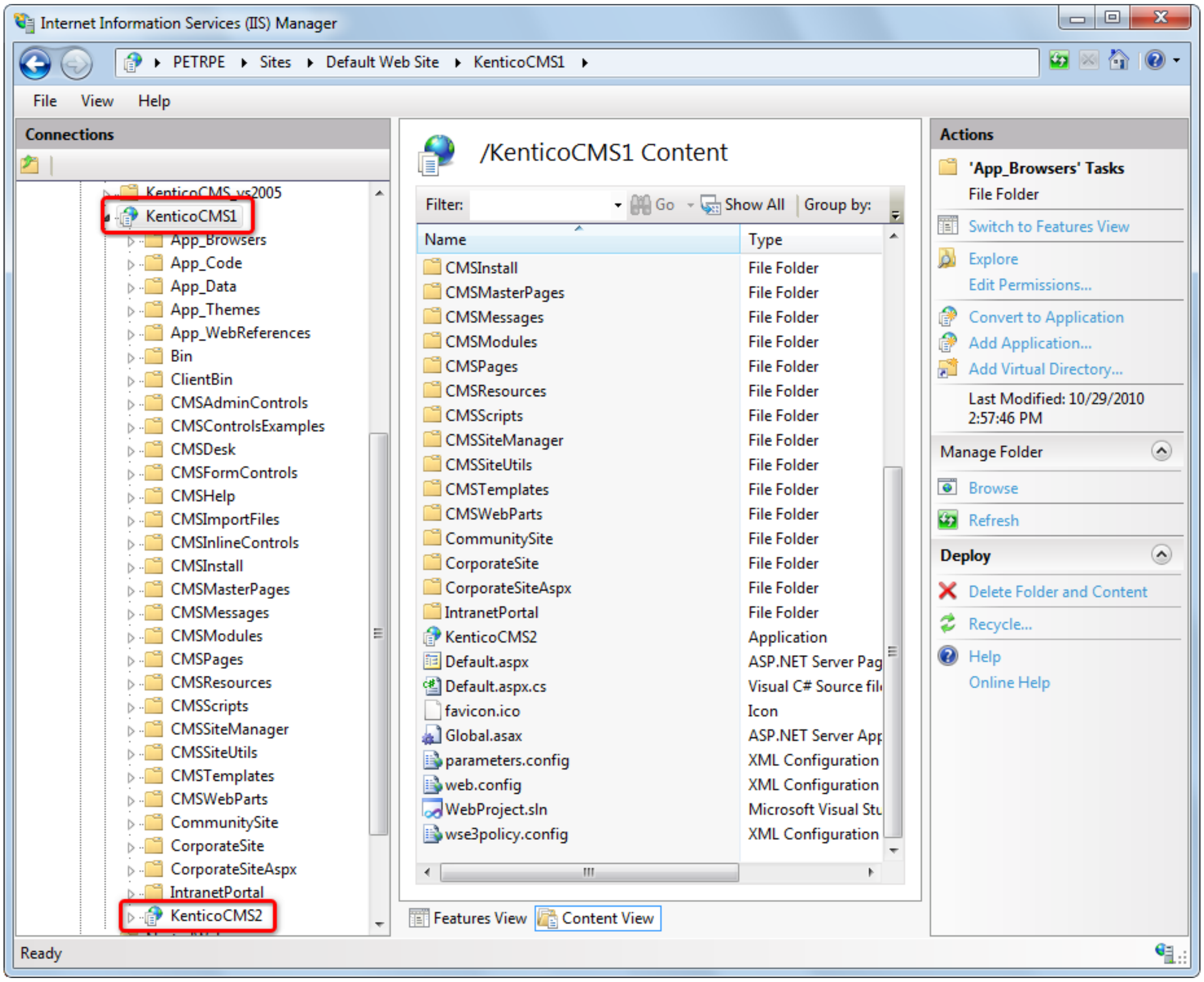
This means that you have two completely independent projects, as you can see in the screenshot below:



2. Open IIS management console and create two applications under the *Default website*:

- **[Default website]/KenticoCMS1** (pointing to physical directory Web1)
- **[Default website]/KenticoCMS1/KenticoCMS2** (child directory pointing to physical directory Web2)

Like this, you have two independent projects configured as nested only in the IIS.

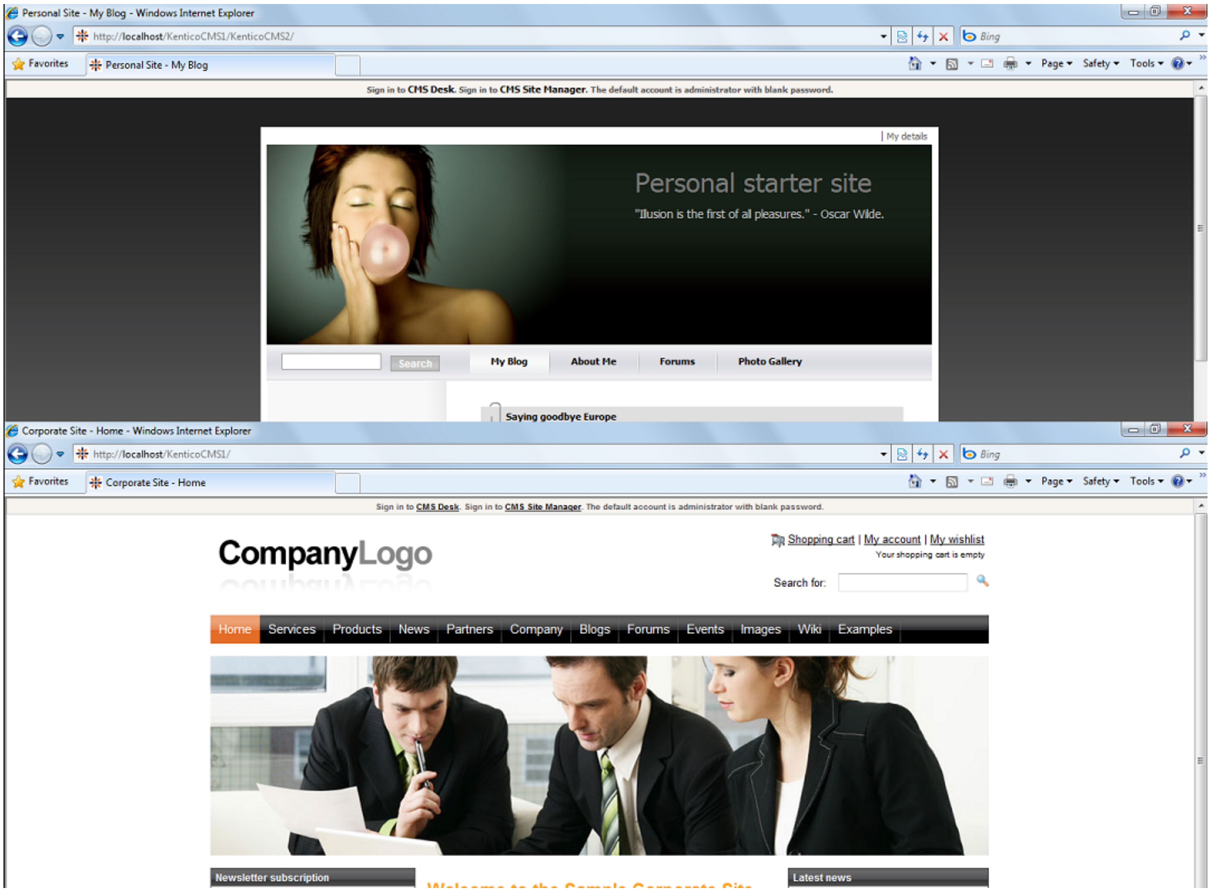


3. The last step is to avoid duplicate module and handler definition keys in the two projects' *web.config* files. To achieve this, open the nested application's (*Web2*) *web.config* file and remove all module and handler definitions from the two sections highlighted in the following code extract.

In case that you needed any additional custom keys in these sections, please ensure that they are not duplicate in the two *web.config* files, i.e. that they are only added in one of the two files.

```
<system.webServer>
  <validation validateIntegratedModeConfiguration="false" />
  <modules>

    // remove or comment out all keys in this section

  </modules>
  <handlers>

    // remove or comment out all keys in this section

  </handlers>
</system.webServer>
```

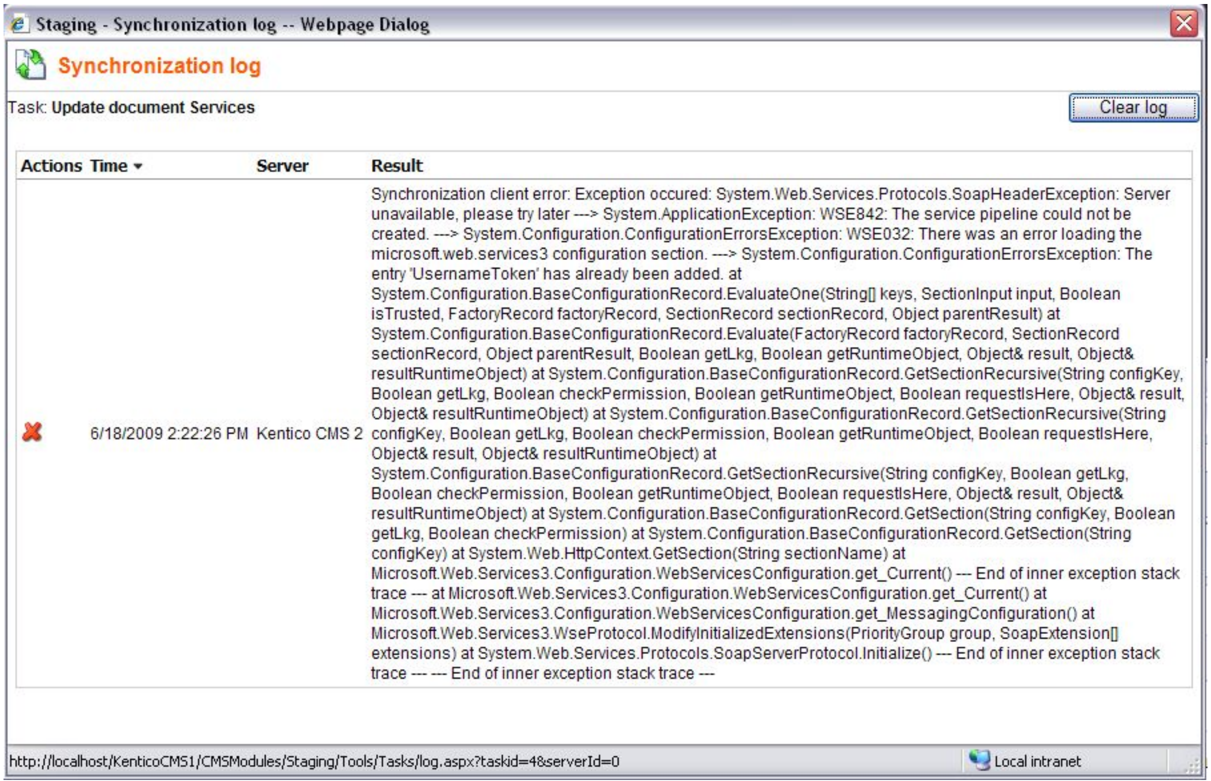
4. The websites are now accessible via nested URLs as you can see in the screenshot below. The websites can be configured independently without experiencing any issues.



Additional configuration for Content staging

You may come to the point where you want to set up Content staging from one of these sites to another. Content staging has some sections in the *web.config* that collide. Config files are inherited within the IIS virtual directories structure (even when the projects are not nested on the file system), but you cannot have the same section of *web.config* twice in the config file.

So if we configure staging from the **KenticoCMS1** site to the **KenticoCMS1/KenticoCMS2** site (see details on how to configure the staging here), the inner project may have issues with the configuration. You would get an error like this:



The imporant part of the error message is "**The username token has already been added**", that means that some of the **configuration is duplicate**.

User name token authentication is defined in the policy file which is referenced from the **<microsoft.web.services3>** section, so **remove the whole <microsoft.web.services3> section** from the **web.config** of the **inner project (Web2)**. Do not remove it from the Web1 (outer) project since this configuration will be used for both websites. After doing this, Content staging should work flawlessly between the sites.

5.13 API Examples

You can use Kentico CMS API to create new websites or update and delete existing ones. To create a new website, you need to import some of the website templates (typically the Blank website template stored in folder App_Data\Templates\BlankSite.

You can find API examples in the following chapters:

- Creating a new website
- Import and export of a website
- Getting website data
- Updating website properties
- Starting and stopping websites
- Deleting websites.

Part VI

Website settings

6 Website settings

6.1 Settings overview

Kentico CMS settings are stored in two places:

- **Site settings** are used for site configuration. These settings are stored in the database.
- **System settings** are used for low-level configuration of the CMS system. These settings are stored in the *web.config* file. You can find more details in Appendix C - Web.config parameters.

Site settings

The site settings are stored in the Kentico CMS database. You can configure them in **Site Manager -> Settings**. There are two types of settings:

- **Global** - applied on all sites in the system
- **Site-specific** - you can edit them after you choose the relevant website in the drop-down list in the upper left corner

The site-specific settings may inherit global values:

Settings are organized in the following categories:

- **Web site** - general website settings
- **Content management** - settings related to content management of your website
- **Files** - settings related to the way files are handled by the system; more info here
- **Membership** - settings related to user accounts, on-line users monitoring, etc.
- **Security** - settings related to passwords, permissions, secured protocols, banned IPs etc.
- **URLs and SEO** - settings related to URLs and SEO, URL extensions, forbidden characters in URLs, permanent redirection etc.
- **Output filter** - settings of how the output HTML code is modified before it is sent to the browser; more info here
- **System** - general system settings, including time zones, UI, DB, event log, notification e-mails, etc.
- **E-mails** - settings related to the e-mail engine, used SMTP server, etc.; more info here
- **Avatars** - settings related to user and group avatar images and their resizing; more info here
- **Bad words** - settings related to the Bad words module; more info here
- **Blogs** - settings related to the Blogs module; more info here
- **Booking system** - settings of the booking system notification e-mails
- **Community** - settings related to the community features - groups, members, activity points, etc.

- **MetaWeblog API** - settings related to MetaWeblog API; more info here
- **Content staging** - settings related to the Content staging module; more info here
- **Forums** - settings related to the Forums module; more info here
- **Media libraries** - settings related to the Media libraries module; more info here
- **Message boards** - settings related to the Message boards module; more info here
- **Messaging** - settings related to the Messaging module
- **Web analytics** - settings related to the Web analytics module; more info here
- **Windows LiveID** - settings related to LiveID authentication; more info here
- **E-commerce** - settings related to the E-commerce module; more info in this chapter of the E-commerce guide
- **OpenID** - setting related to OpenID; more info here.
- **Payment gateway - Authorize.NET** - settings of the Authorize.NET payment gateway used by the E-commerce module; more info in this chapter of the E-commerce guide
- **Facebook Connect** - settings related to connecting to Facebook; more info here
- **Payment gateway - PayPal** - settings of the PayPal payment gateway used by the E-commerce module; more info in this chapter of the E-commerce guide
- **Microsoft SharePoint** - settings related to LiveID authentication; more info here
- **Intranet portal** - settings related to Intranet portal
- **Project management** - settings related to the Project management module; more info here
- **WebDAV** - setting related to the WebDAV integration module; more info here

You can find more details on particular settings keys when you mouse-over the key name - you will see a pop-up info box that displays the key description. Detailed descriptions can also be found in **Context help**, which can be displayed by clicking the icon at the top right corner of the page.

Part VII

Development

7 Development

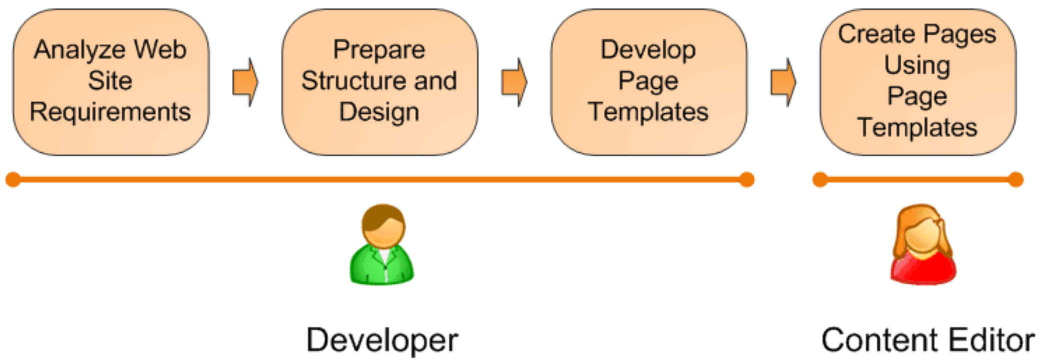
7.1 Overview

This section provides reference on various topics related to website development and system administration. Open the particular sub-chapters in order to get the related topics displayed.

7.2 Web development overview

7.2.1 The role of a web developer

The role of a web developer is described in the following figure:



This figure shows how you develop a website and how roles are split between developers and editors. A typical development process consists of the following steps:

1. The developer analyzes client requirements.

2. The developer prepares site structure (site map) and web design.

3. The developer creates **page templates** for every type of page required on the site (home page, solutions, products, news, etc.)

4. The content editor creates new pages - they enter text and images into the page templates defined by the developer.

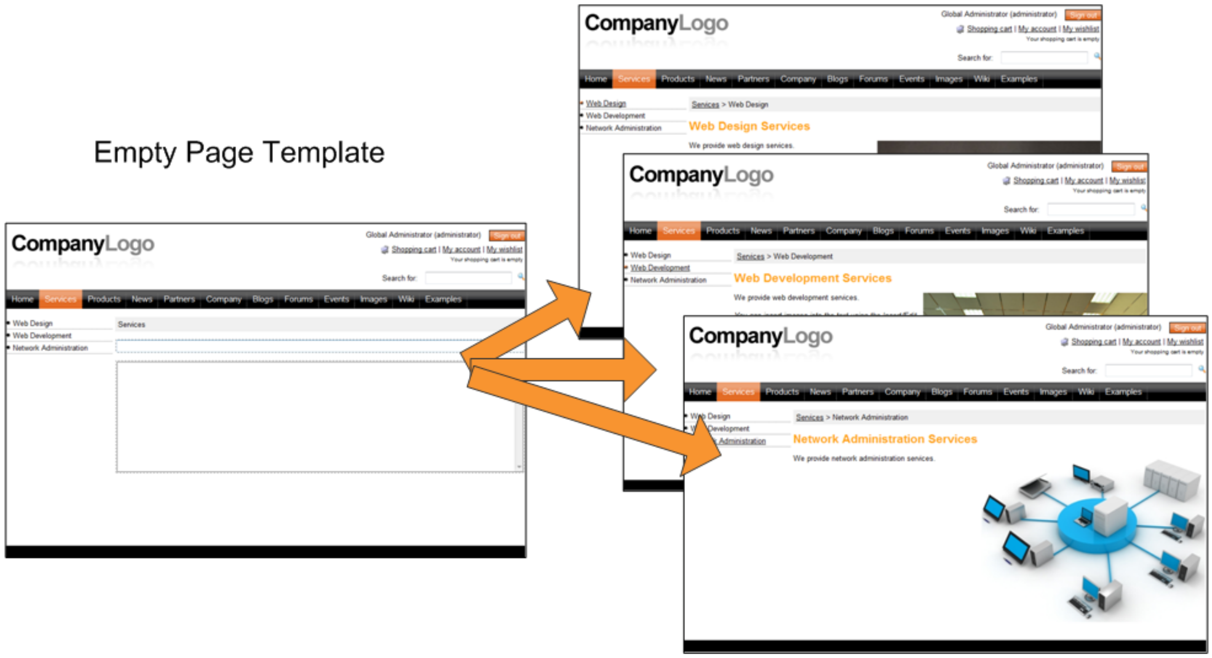
7.2.2 What is a page template

A page template is a predefined look of the page that allows content editors to enter the content. A single page template can be re-used for multiple pages with the same structure and design, but with different content.

The templates allow content editors to focus just on content editing, without the need to take care of page formatting. They also help keep the web design consistent throughout the whole website. The following figure shows how a single page template can be used for multiple pages:

Pages Created Using the Page Template

Empty Page Template



The following topic describes the two development models for creating such page templates with Kentico CMS.

7.2.3 Portal templates versus ASPX templates

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

- **Portal Engine** - this model allows you to build websites using a portal engine. It's the recommended way for most developers since it doesn't require programming in Visual Studio. Instead, you can simply 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 websites using standard ASP.NET architecture and using standard development tools, such as **Visual Studio**. This model requires you to be familiar with ASP.NET development and have at least basic programming knowledge of C# or VB.NET.

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

The following table compares portal engine and ASPX templates:

	Portal Engine	ASPX Template
How you work	You build the website using the browser-based interface. No programming knowledge is required for common tasks.	You build ASPX pages in Visual Studio. At least basic programming knowledge of ASP.NET and either C# or VB.NET

		is required.
How you assemble pages	You use built-in or custom web parts that you place into customizable page layouts (HTML code with placeholders for web parts).	You use built-in or custom ASP.NET server controls that are placed on the ASPX pages. You can also use the code behind files - these are standard ASPX pages and they are part of the website project that you can open in Visual Studio. You can also place web parts (which are actually ASCX user controls) on the page templates if the appropriate server control is not available.
Master pages and visual inheritance	Sub-pages inherit the content from the parent pages by default (so called "visual inheritance"). The inheritance can optionally be broken if you want to create a page without parent content.	All page templates may use a master page, which is a standard ASP.NET 2.0 master page. The pages do not inherit content from their parents, they only inherit content from the master page (if it's used).
Custom code integration and extensibility	You can create your own user controls (ASCX) and web parts (ASCX with specific interface) in Visual Studio if you need to integrate some custom functionality. You can add any custom controls and code to the web parts or user controls that you use on your website.	You build standard ASPX pages with codebehind files, which means you can put any custom controls and code onto the page in Visual Studio as you normally would.
Advantages	• 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 the web browser.	• Standard ASP.NET architecture. • You can use your favorite development tools, such as Visual Studio for all changes.
Disadvantages	• Proprietary architecture and development.	• Requires ASP.NET programming knowledge.

Is Kentico CMS just another portal engine?

Now you may wonder what the difference between Kentico CMS and DotNetNuke or Rainbow Portal is.

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. This means it provides features of advanced CMS systems, such as:

- content repository using a logical tree hierarchy for all types of content - see [Where is the content stored?] for details
- content/design separation
- custom document types with custom fields
- workflow and versioning
- content locking (check-out, check-in)
- multilingual content
- content preview and content staging
- document-level permissions with permission inheritance
- full-text search in all content
- document management features for uploaded files

Moreover, Kentico CMS comes with many **professional and flexible built-in modules out-of-the-box**, including:

- Newsletters
- On-line forms
- Forums
- E-commerce
- Content Staging
- Image gallery
- Blogs
- Polls
- Web analytics
- Reporting

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

7.2.4 Portal engine development model

7.2.4.1 Portal engine overview

Each page in a Kentico CMS website uses some page template. A **page template** consists of a layout with particular web parts and their configuration. The portal engine page templates are physically stored in the database, so you will not find them on the disk.

Design of the page template is defined by the **page layout**, which is a standard HTML code. The HTML code can be used to define two-column, three-column, or virtually any custom layout you can think of.

The page layout contains special markup tags that specify places where developers can place web parts - so called **web part zones**. A web part zone can contain any number of web parts.

A **web part** (also called "servlet", "portlet" or "module" in other solutions) is a piece of code that displays some content or provides some functionality. Technically, web parts are standard ASCX user controls with some predefined programming interface (inherited from *CMSAbstractWebPart*). More information about web parts can be found in the [Development -> Web parts] chapter.

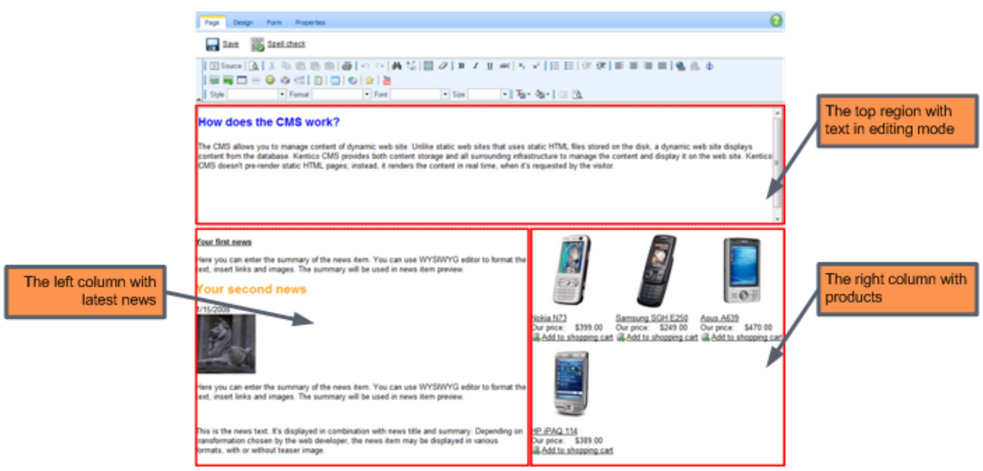
The following figures show an example of a simple page, its page template, layout, web part zones and web parts:

1) The sample page on the live site

This image shows our sample page on the live site. It displays text on the top, news in the left column and products on the right.

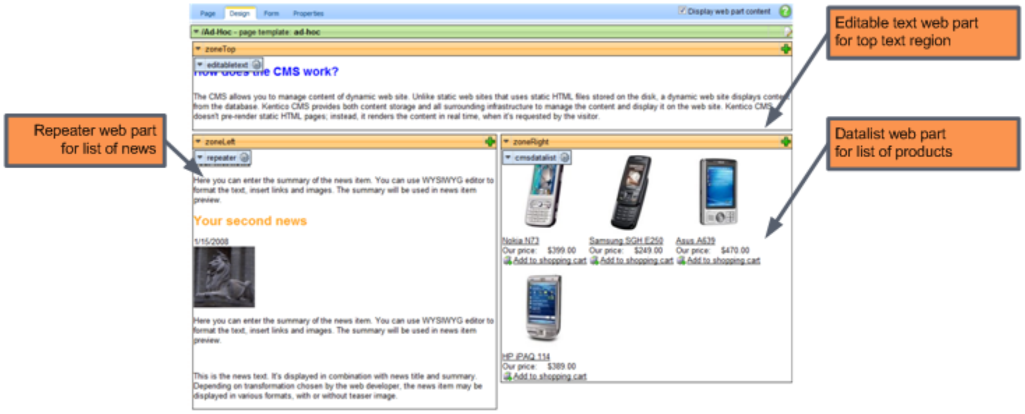
2) The sample page in the editing mode

Here you can see the top region with editable text and the lists of news and products.



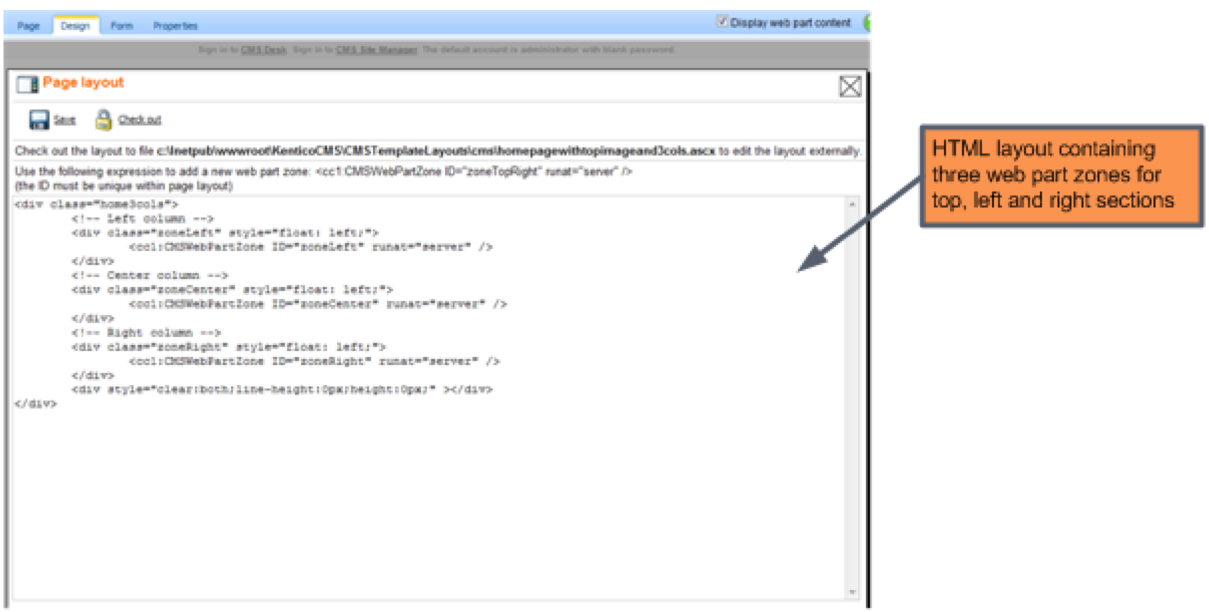
3) The page template of the sample page on the Design tab

This is the **Design** mode view of the page template used for our sample page. As you can see, it consists of three web part zones - **zoneTop**, **zoneLeft** and **zoneRight**. The web part zones contain particular web parts.



4) HTML layout of the page template

This image shows the HTML layout of the page template with tags representing web part zones on the top and in the table columns. We are using tables for the layout here, but you can easily use a CSS-based layout - you have full control over the HTML code and CSS styles.



7.2.4.2 Creating a new page template

There are two ways to create a new page template:

1. Creating a blank page using an ad-hoc template.
2. Cloning an existing page template.

We will start with the first option.

What is an ad-hoc template?

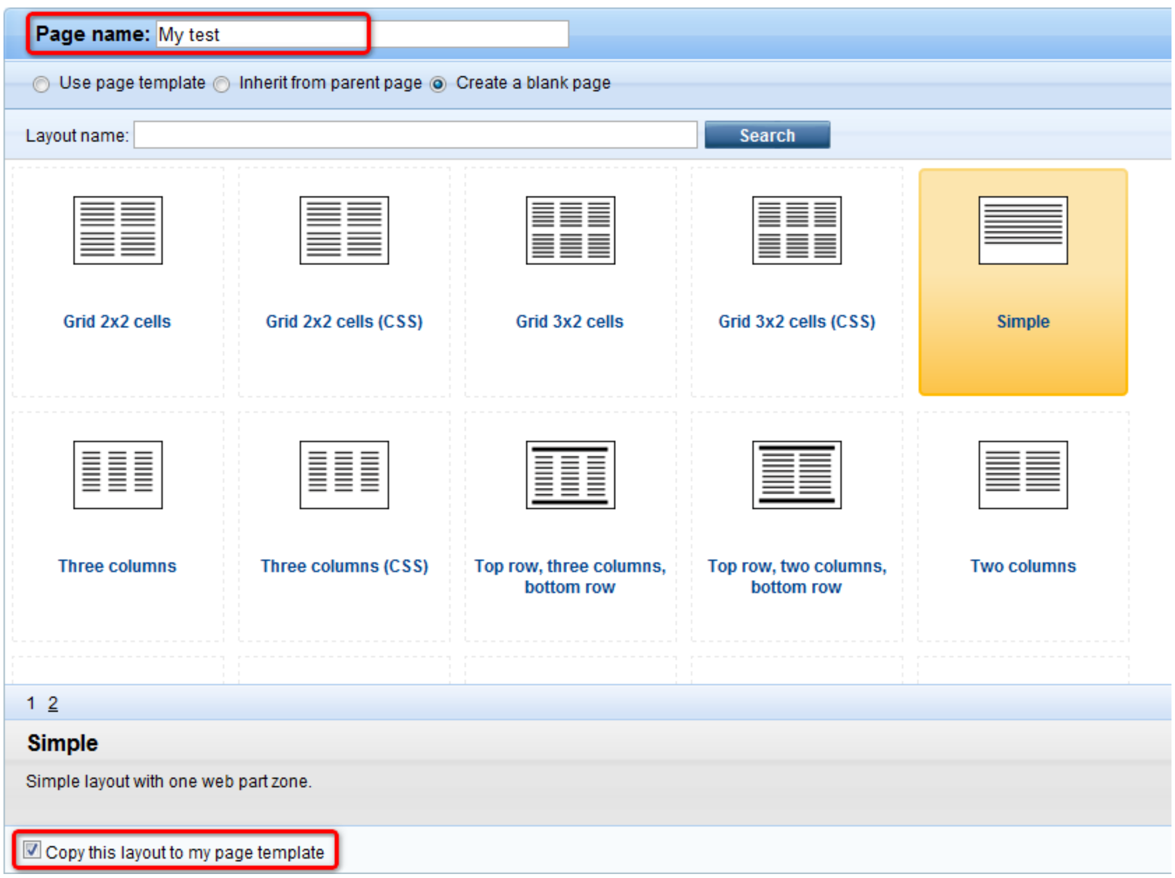
An ad-hoc template is a page template that is used by only one page. An ad-hoc template doesn't have any name. If you want to re-use the template for multiple pages, you need to save it as a named template first.

Creating a blank page using an ad-hoc template

Please note: It's recommended that you use the sample **Corporate Site** for this example.

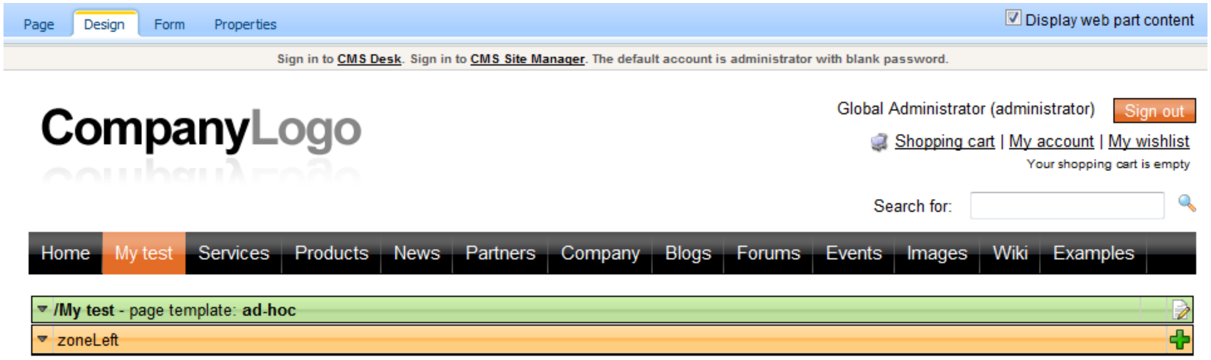
1. Go to **CMS Desk -> Content**, click **New** and choose to create a new **Page (menu item)** under the website root.

2. Enter *My test* into the **Page name** field and choose the **Create a blank page** option. Choose **Simple** from the list of layouts and leave the box **Copy this layout to my page template** checked. Click **Save**.

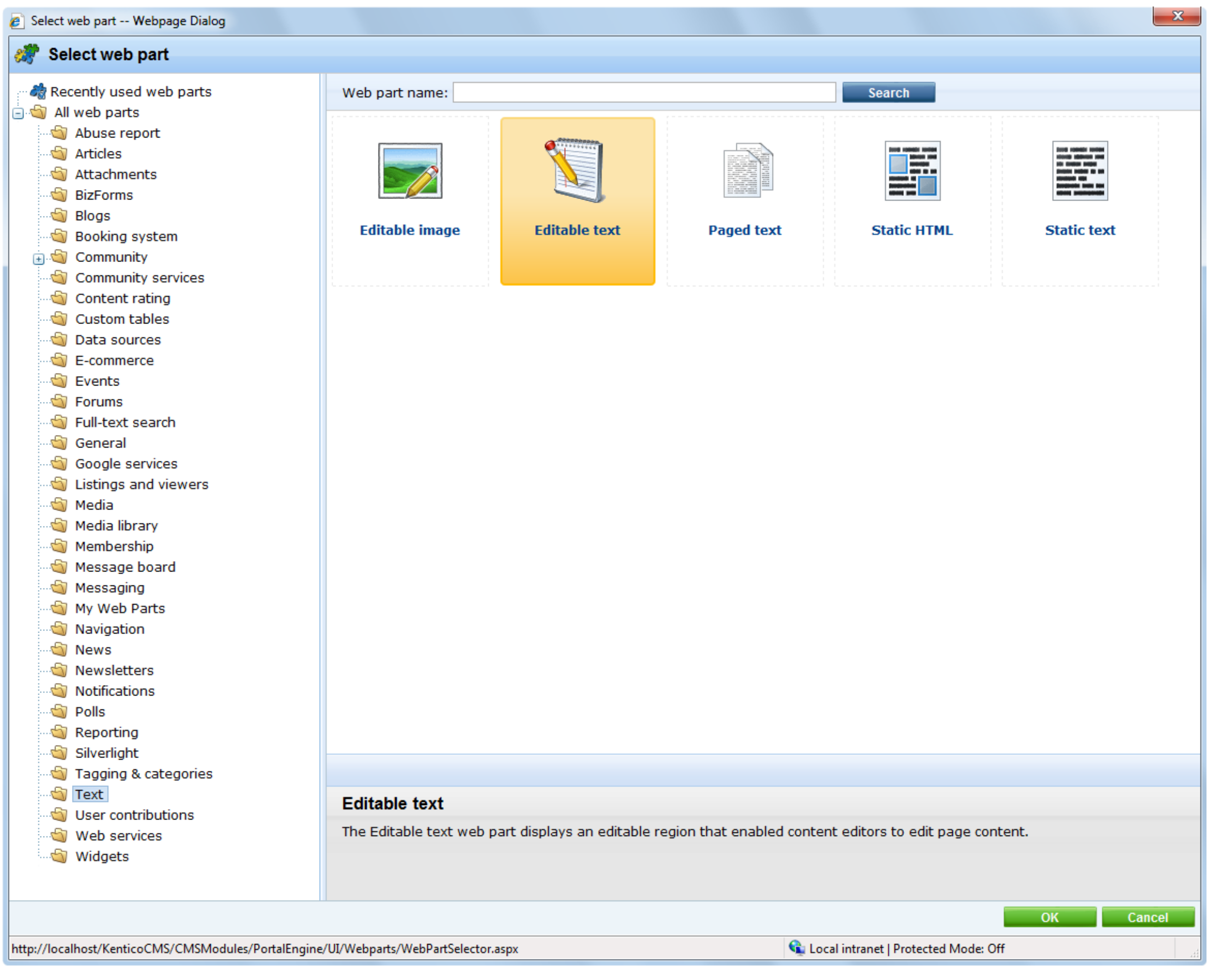


3. The page is now displayed in the tree view and selected. Click the **Design** tab. You will see a page as in the screenshot below.

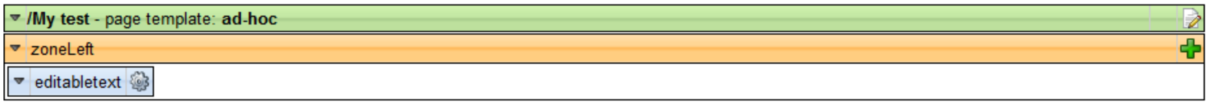
The green box with **/My test** title displays the current page. As you can see, we are now using an ad-hoc template that is specific for this page. The yellow box with the **zoneLeft** title displays the web part zone, where web parts can be placed.



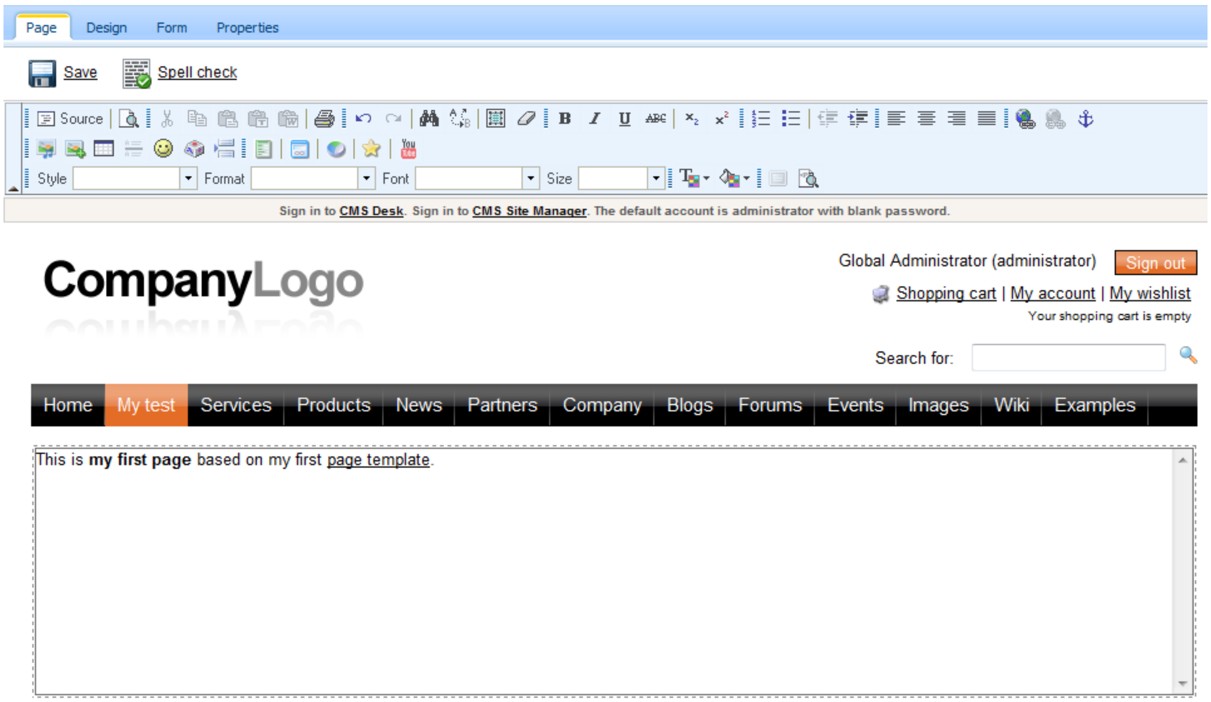
4. Click **Add web part** (+). The **Select web part** dialog opens. Select the **Text/Editable text** web part and click **OK**. This web part allows you to create editable regions where content editors can insert content.



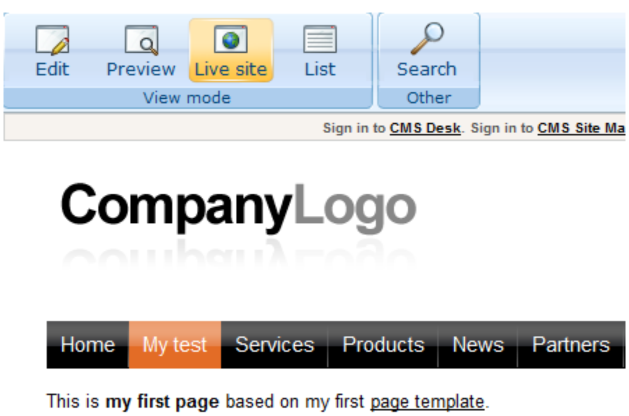
5. The **Web part properties** dialog opens. Enter *Main text* into the **Editable region title** property and click **OK**. The web part is now added to the web part zone:



6. Now click the **Page** tab. You will see the page in editing mode. The page now contains your web part which displays the editable region. Enter some text into the region and click **Save**.



7. Click **Live site** in the main toolbar. You will see the live version of the page that is displayed to the visitors:



What you did

You have created a new page based on an ad-hoc page template. The page template defines the page structure and contains the editable text web part which enables content editors to enter text into the page.

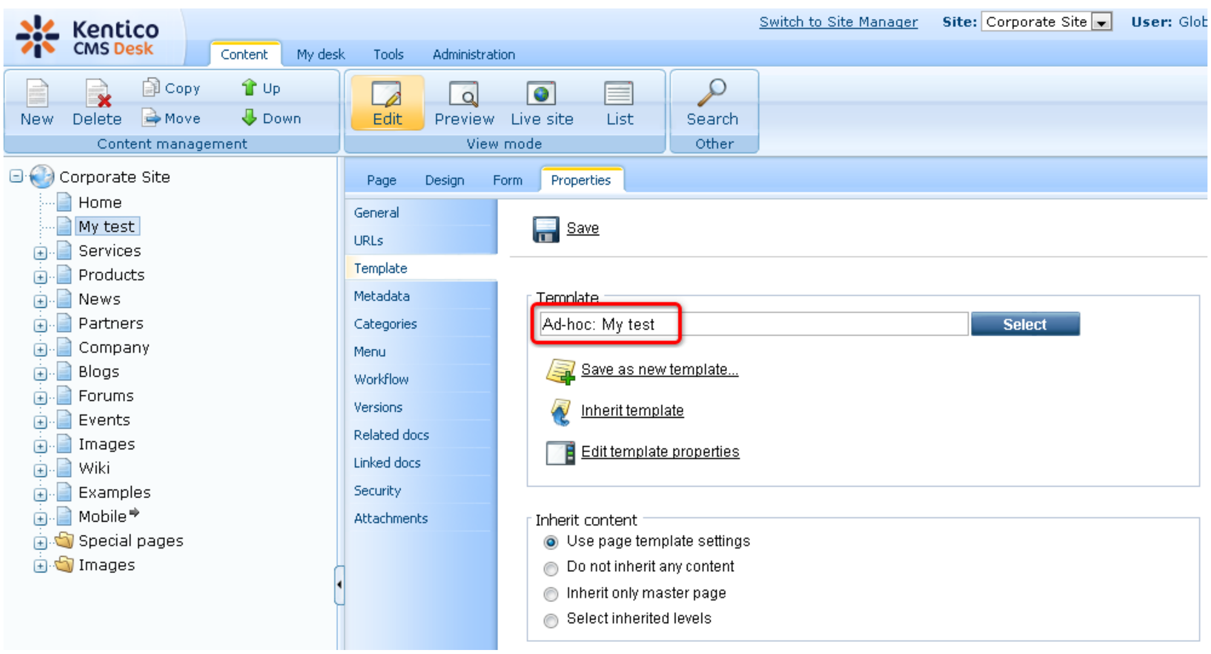
Important!

Please, keep in mind that when you create a page based on an ad-hoc page template and later delete the page, the corresponding ad-hoc page template is deleted as well.

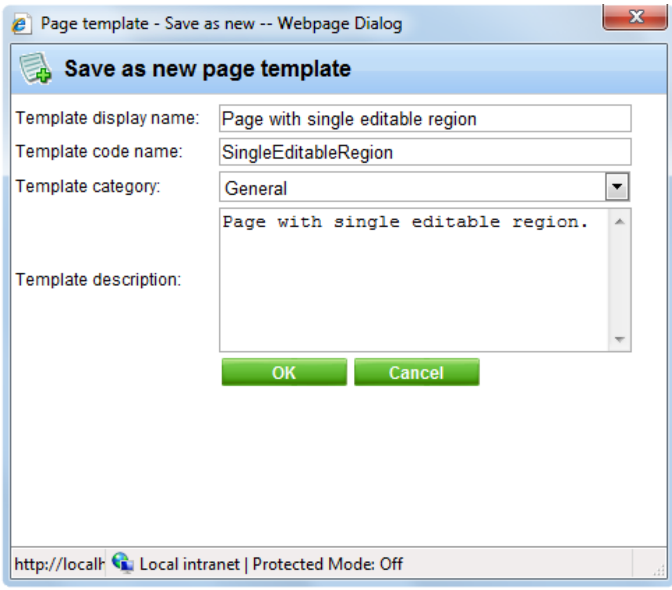
7.2.4.3 Re-using an ad-hoc page template

In many cases, you may want to re-use an ad-hoc page template for other pages. In this case, you need to **save the ad-hoc template as a new re-usable template**.

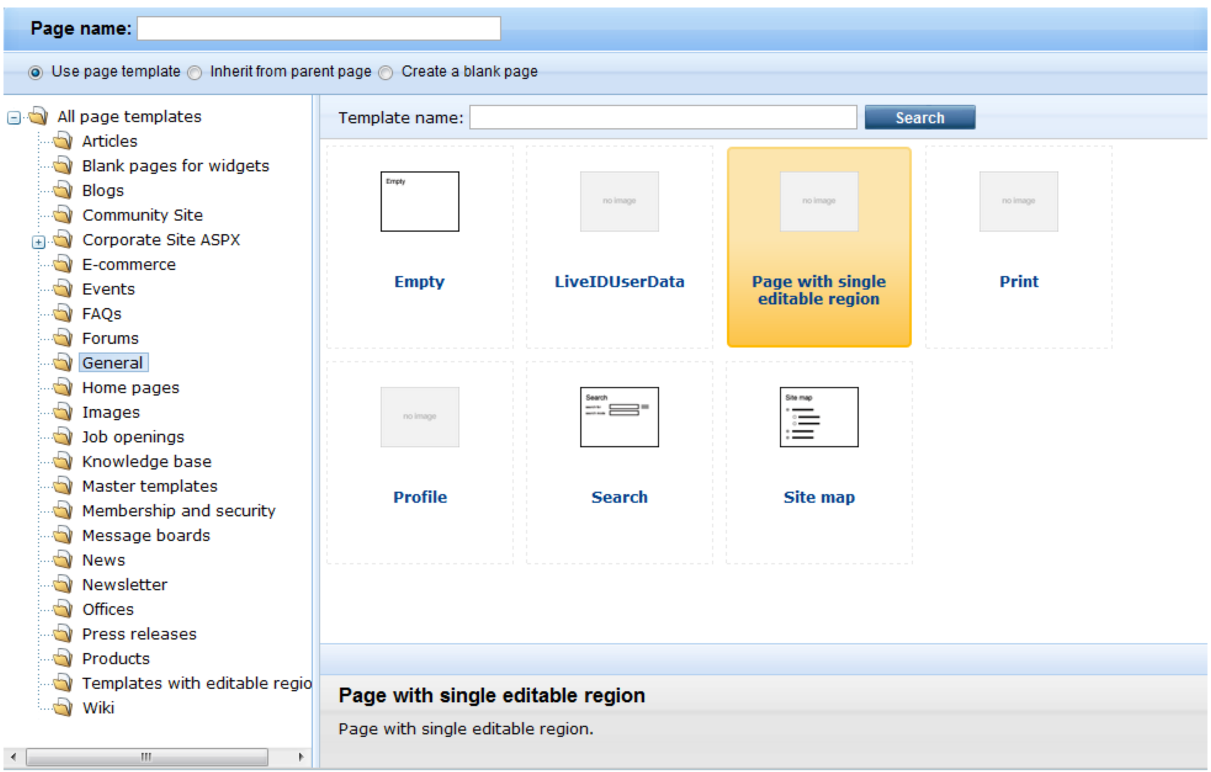
1. Go to **Content ->** switch to **Edit** mode on the main toolbar **->** select your new page created in the previous example **-> Properties -> Template**. You will see a page as in the following screenshot. As you can see, the current page is based on an ad-hoc template.



2. If you want to re-use this page template for some other page, click **Save as new template**. The **Save as new page template** dialog opens. Fill in values as shown in the picture below and click **OK**, then click **Save** on the **Template** dialog.



3. Now when you try to create a new page, you will be able to create the page based on this new page template:



Additionally, the template is now added to the content tree at **Site Manager -> Development -> Page templates**, where it can be configured.

What you did

You have converted the previously created ad-hoc page template that was specific only for a single page into a re-usable page template that can be used for any number of pages.

7.2.4.4 Page template scopes

Page template scopes define where a page template can be used. **Important - template scopes are not a security feature!** They only provide a way how to simplify the user experience by not offering too many page templates to the users!

This means that you can limit which page templates will be offered to the users in the page template catalog in a certain location. But when a user creates a page there, they can move it to another location, even to one which is not allowed by the template scope.

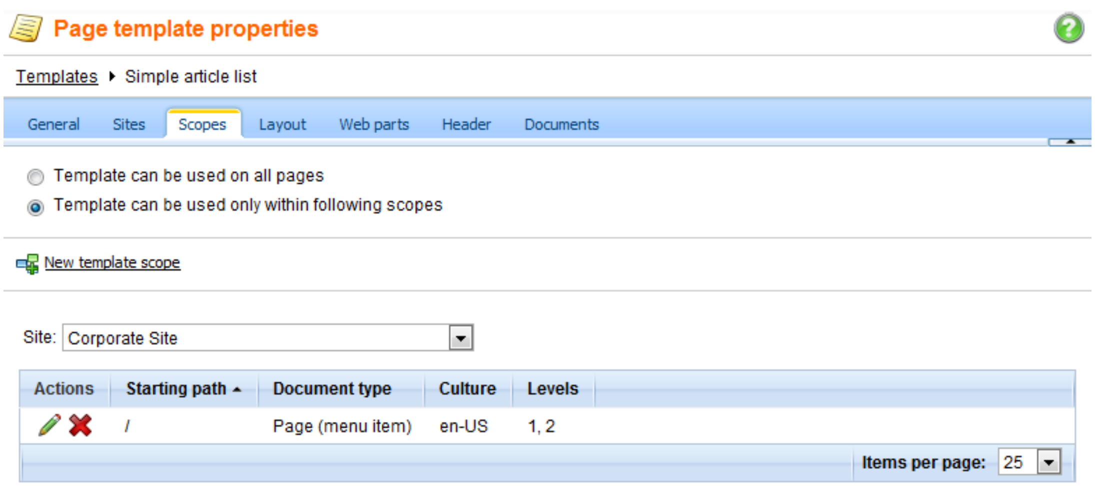
Template scopes can be defined on the **Scopes** tab when selecting a page template from the content tree at **Site Manager -> Development -> Page templates**. You have the following two basic options:

- **Template can be used on all pages**: the use of the page template is not limited and it can be used anywhere
- **Template can be used only within following scopes**: if selected, the page can be used based on the defined template scopes

If you choose the second of the options mentioned above, UI controls for managing the scopes will be displayed below the line.

Based on the site selected in the **Site** drop-down list, you can display either site-related scopes or global scopes. Site-related scopes are applied only on the selected site, while the **(global scopes)** option indicates that the scopes are valid on all sites in the system.

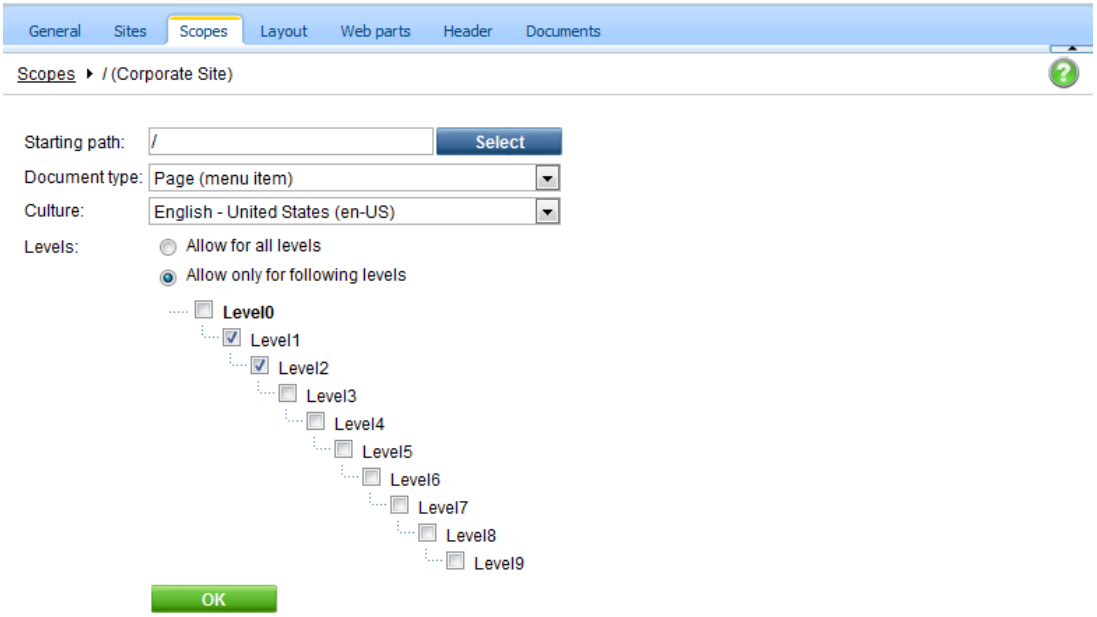
Using the **New template scope** link, you can create new template scopes. Listed scopes can be **Edited** () or **Deleted** ().



When creating a new scope or editing an existing one, you can specify the following details, which define where the page template can be used:

- **Starting path**: path in the content tree where the page template can be used; the page template cannot be used outside this path
- **Document type**: the page template can be used only for documents of the specified type

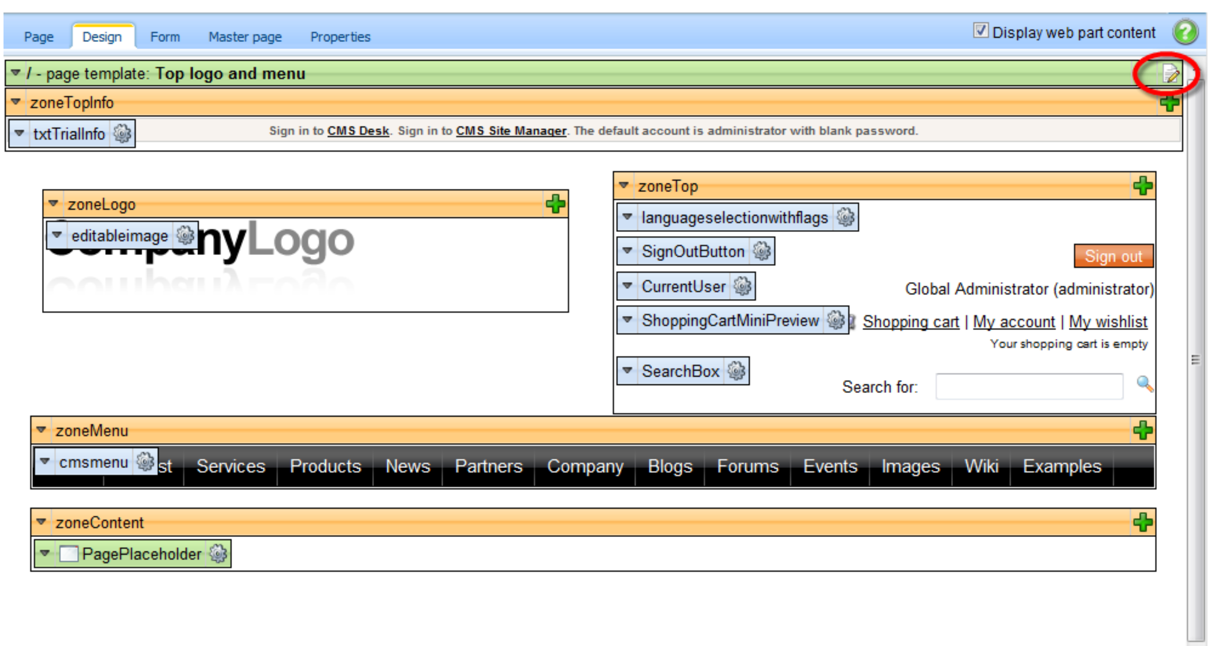
- **Culture**: the page template can be used only in the specified culture
- **Levels**: the page can either be used on all levels or only on the selected levels in the content tree

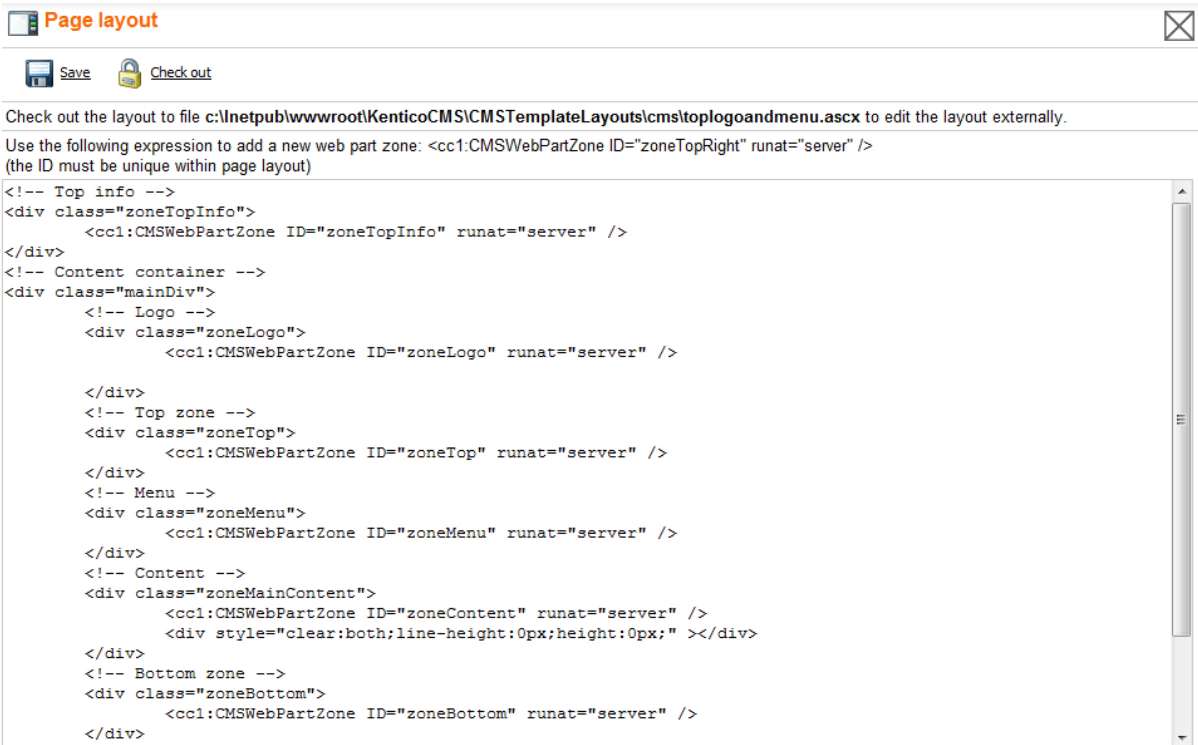


7.2.4.5 Page layouts

A **page layout** consists of the **HTML layout** of the page template and **web part zones** that specify regions where web parts can be placed. You will use page layouts to define the **layout and design of your site**.

The easiest way to edit a page layout is to switch to the **Design tab** of the page and click the **Edit layout** ( ) button at the top-right:





The web part zones are defined by the following tag:

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

The ID value must be unique within the given layout.

As you can see, the layout code is standard HTML. It means **you have full control over rendered HTML code** and you can choose between table and CSS layout.

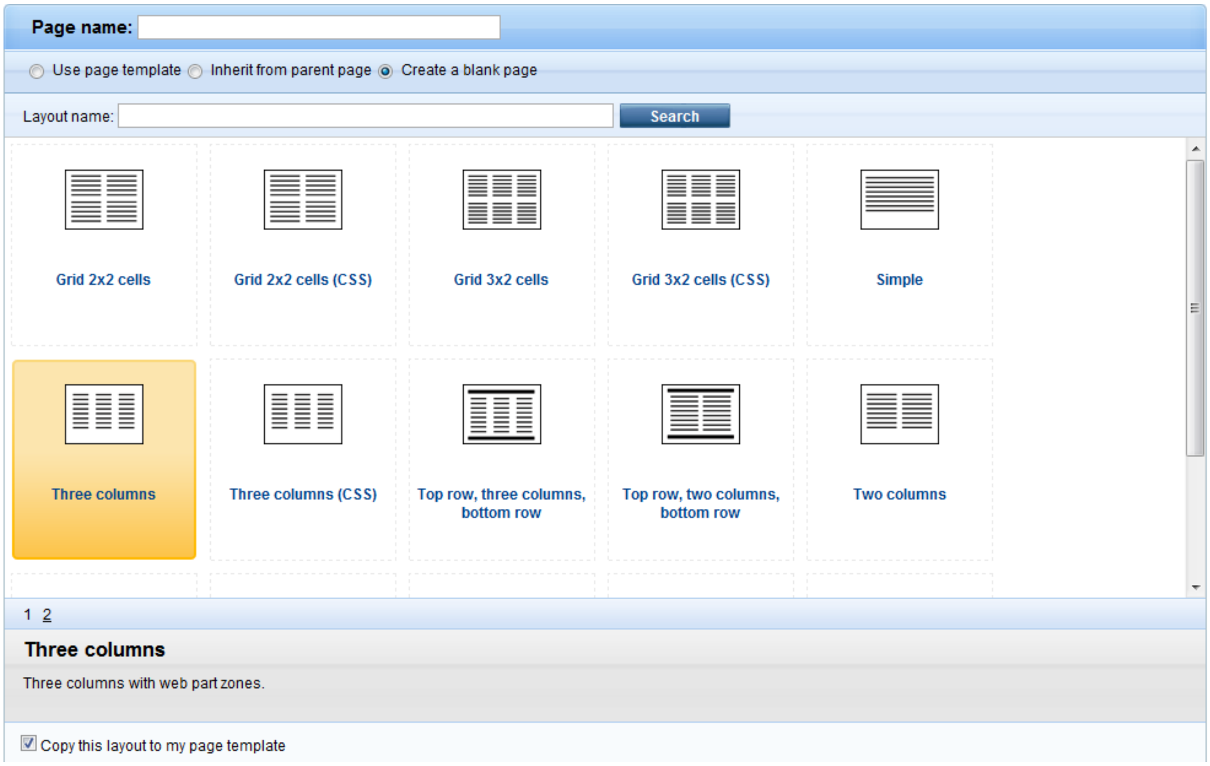
Editing the layout in your favorite editor

Editing of HTML code in a plain text area may not be very comfortable. However, you can use the **Check out to file** button to save the layout code to the disk and open it in your favorite HTML editor.

Please note: The file is saved on the disk where the web application is running.

Managing pre-defined page layouts

When you are creating a new blank page, you can choose from pre-defined page layouts:



The **Copy this layout to my page template** option allows you to decide whether you will be using the shared layout or if you create a copy of the layout specific for your page template only. If you choose the **shared layout** and make changes to the HTML layout code, they will affect all pages that use the same shared page layout, so it's better to use a copy of the layout in most cases.

You can manage the pre-defined page layouts in **Site Manager -> Development -> Page layouts**. You can configure the following properties:

Layout display name	The name of the layout displayed to the users.
Layout code name	The name of the layout used in the code.
Layout description	Optional description.
Page layout thumbnail	Upload field for the layout preview image (available only when editing an existing layout). This image is used in the page layout selection dialog shown when creating a new blank page as seen in the image above.
Layout code	ASCX code of the page layout.

Sample layout code

The following sample page layout code uses a table to define a two-column layout:

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

```
      <cc1:CMSWebPartZone ID="zoneLeft" runat="server" />
    </td>
    <td>
      <cc1:CMSWebPartZone ID="zoneRight" runat="server" />
    </td>
  </tr>
</table>
```

The following layout code defines the same two-column layout, but using DIV elements and CSS styles:

```
<div style="width: 50%;">
  <div style="width: 80%; float: left;">
    <cc1:CMSWebPartZone ID="zoneLeft" runat="server" />
  </div>
  <div style="width: 50%; float: right;">
    <cc1:CMSWebPartZone ID="zoneRight" runat="server" />
  </div>
</div>
```

7.2.4.6 The master page concept

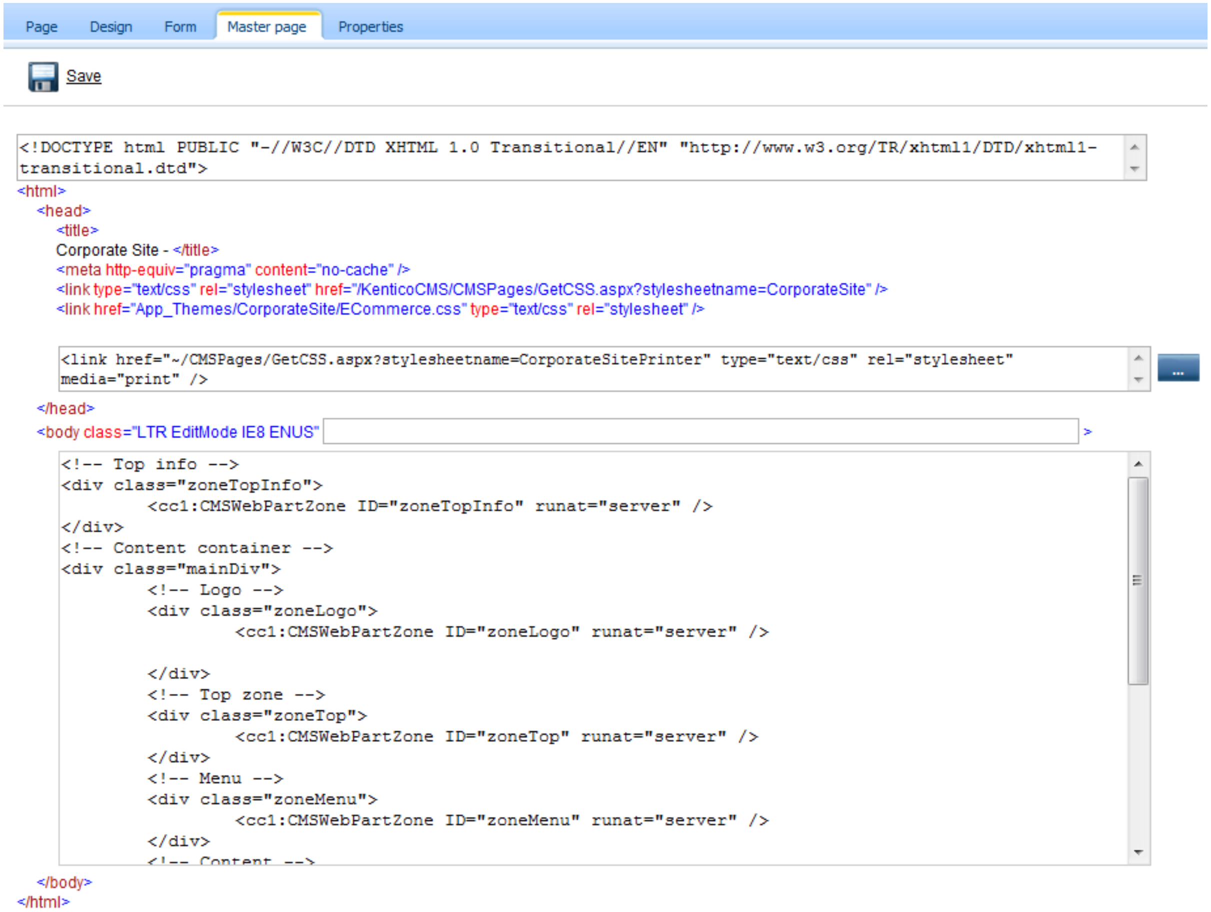
Master pages represent the powerful concept of sharing the same header and footer for all pages on the website. It allows you to manage repeated items, such as the site logo, main menu and footer content in a single place.

The root of the content tree is always a master page. You can also configure any other page template to be a master page by enabling the **Master page** option on the **General** tab of the page template editing interface at **Site Manager -> Development -> Page templates**.

The following figure shows how the same master page is used for the home page and product page. As you can see, the pages are inserted inside the master page:

7.2.4.7 Editing the master page

Master pages are either the root of the content tree or pages whose page template has the **Master template** option enabled on the **General** tab of the page template editing interface at **Site Manager -> Development -> Page templates**. They can be edited just like any other page. You can use the **Design** tab to edit the web parts and HTML layout of the master page. In addition, there's a special **Master page** tab available only for master pages:



As you can see, you can define the following sections of the HTML code of the master page. This code is used for all pages that inherit content from the master page:

- **DOCTYPE -** here you can enter any code that needs to be placed at the beginning of the file, typically the DOCTYPE definition
- **HEAD -** here you can put any HTML code that needs to be placed inside the **<head>** element of the page.
- **BODY -** here you can add custom attributes to the **<body>** element.
- **Master page code -** this is actually the HTML layout of the master page template. This is the same code that you can edit on the **Design** tab, by using the **Edit layout** ( ) button.

Please be aware that the code between the editable sections is of informative character only and may not be identical to the actual rendered code.

Page placeholder

A master page must always contain the **Page placeholder** web part that specifies where the content of sub-pages should be loaded. Visual inheritance is described in more detail in the next topic.

7.2.4.8 Visual inheritance

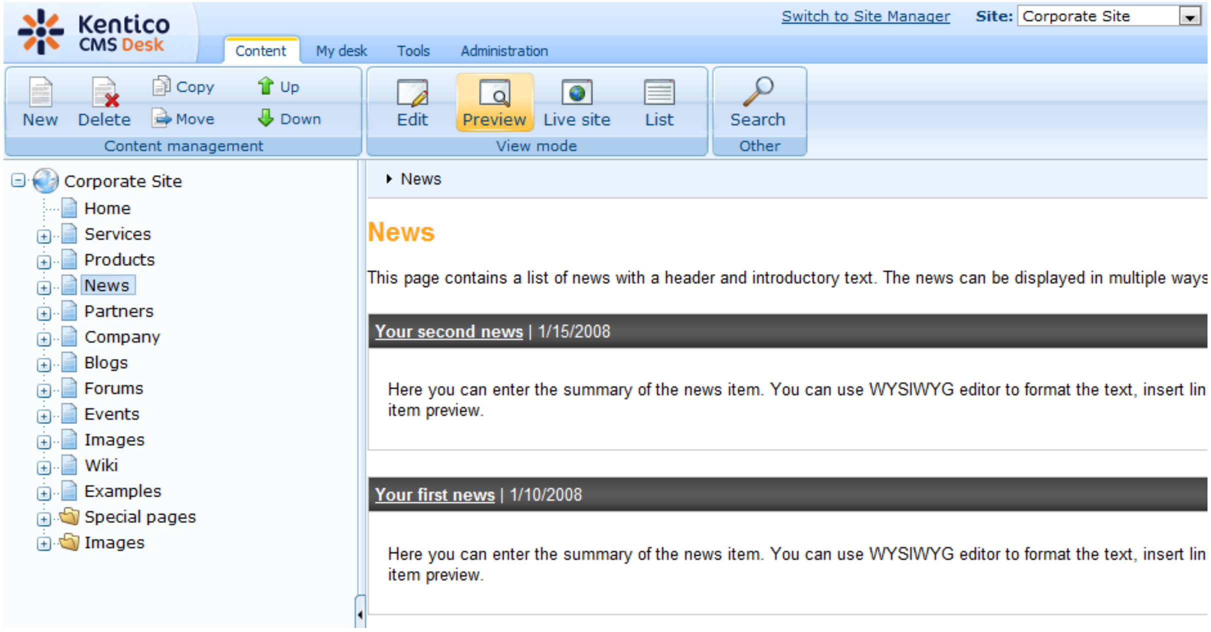
As you can see in the topic The master page concept, the content of sub-pages is displayed within the master page or generally within any parent page using the **Page placeholder** web part. The result of this approach is that the sub-page content is "nested" inside the content of the parent pages.

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

Using the "Inherit content" property of the page template

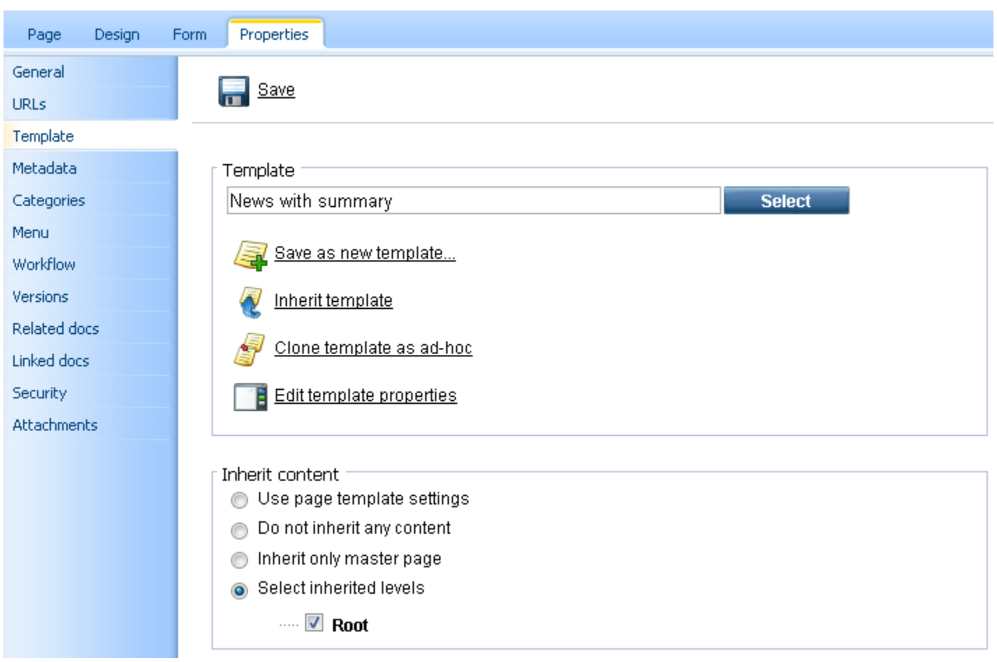
Click the **/News** page and click **Properties -> Template -> Edit template properties**. Now you can set the **Inherit content** value to **Selected inherited levels** and check only the **Level1** box. It means that only the content from the first level of the content hierarchy will be displayed and the master page (root) is not inherited. Click **OK** to save the changes.

The page will now look like this:



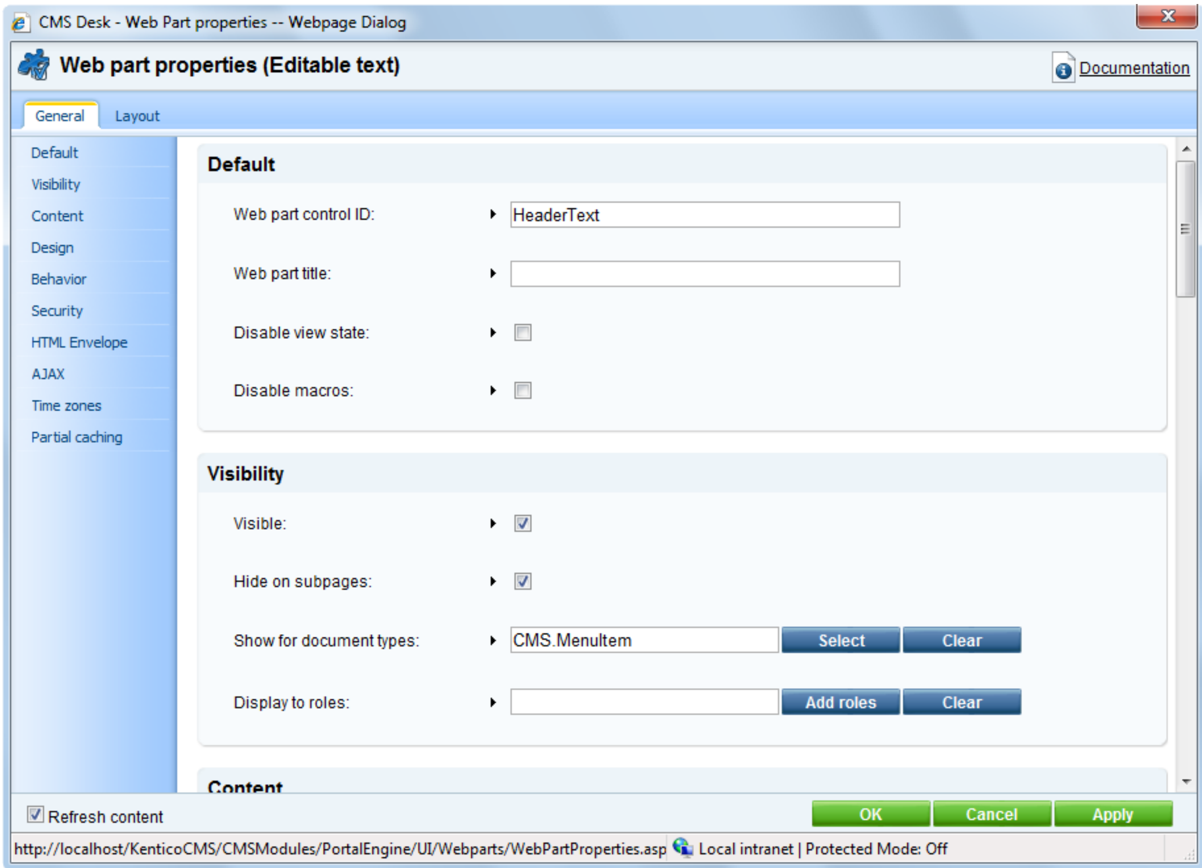
Set the **Inherit content** value back to **Level0** for now.

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 the page override the page template settings:

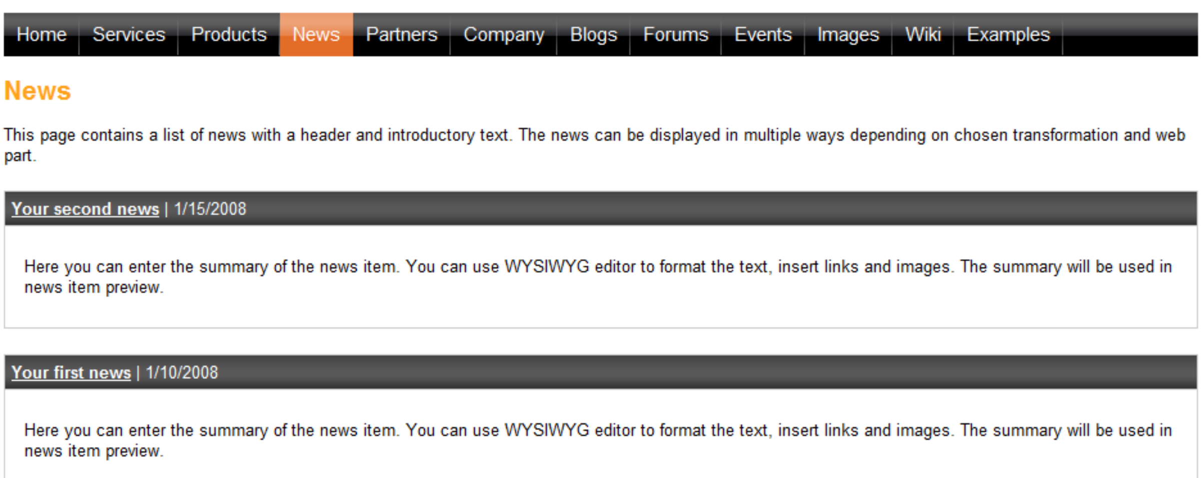


Using the "Hide on sub-pages" web part property

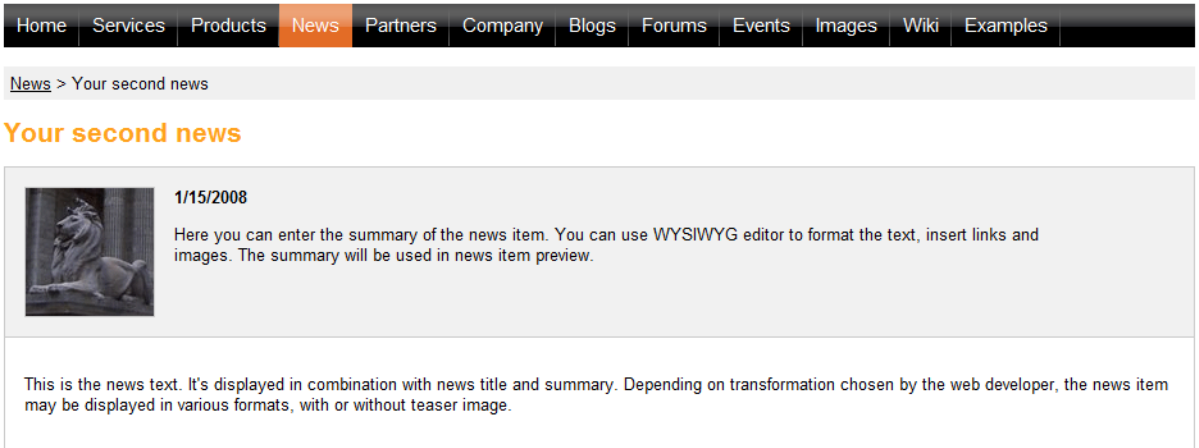
Click **/News**, switch to the **Design** tab and click the **Configure** ( ) button of the **HeaderText** web part. The web part has the property **Hide on sub-pages** set to true:



Click **Cancel** and click **Live site**. Please note that when you display the list of news, the title **News** is displayed:



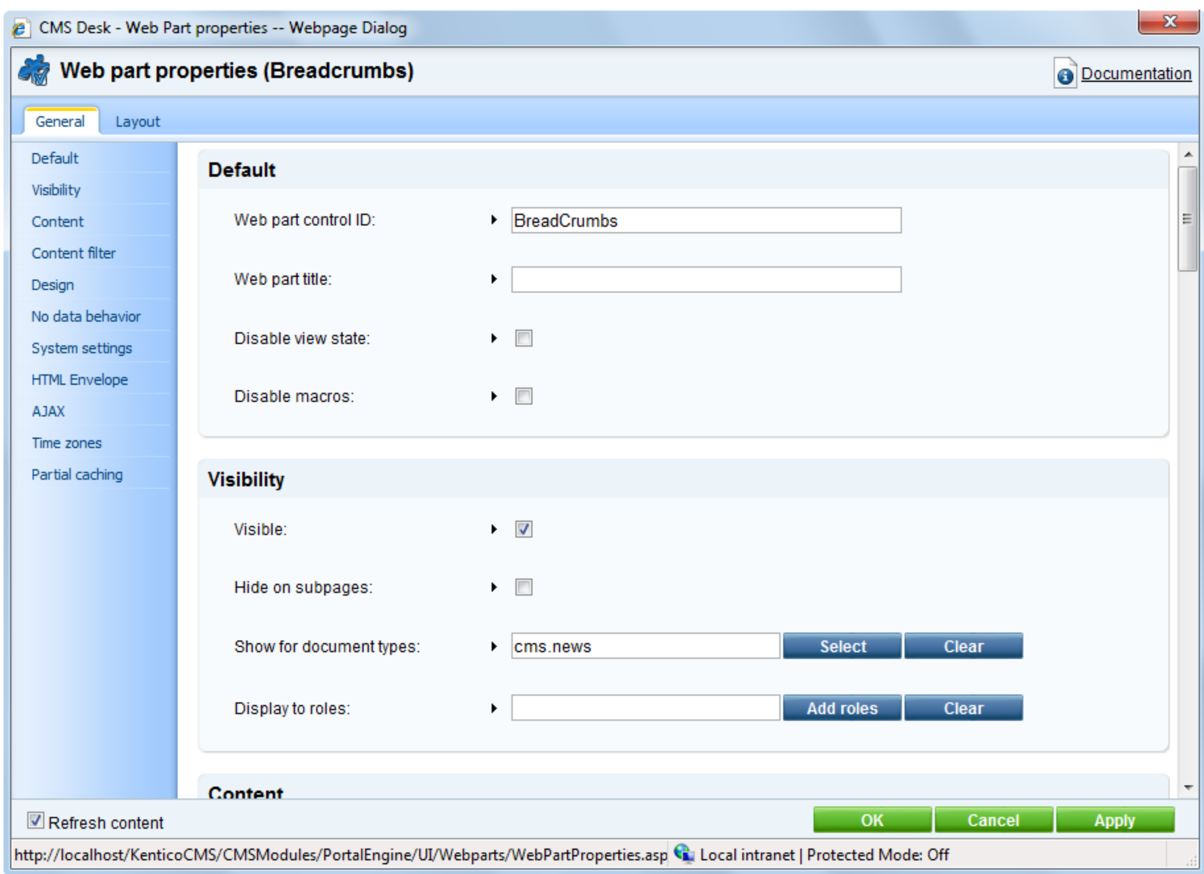
If you go to some particular news item, the title is hidden:



This is ensured by the **Hide on sub-pages** property that hides the web part displaying the **News** title.

Using the "Show for document types" web part property

The **Show for document types** property allows you to define a list of document types for which the web part will be displayed. Click **/News**, click **Design** and click **configure** in the **BreadCrumbs** web part. The **Show for document types** field is set to the **News** document type. Click **Cancel**.



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

When you go to some particular news item, the currently selected document is a news item and the breadcrumb navigation is displayed:

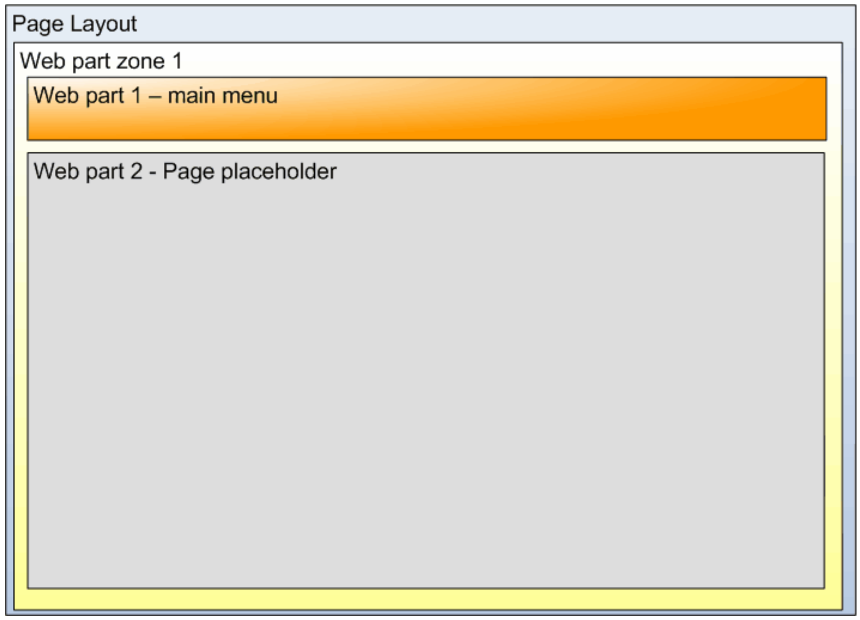
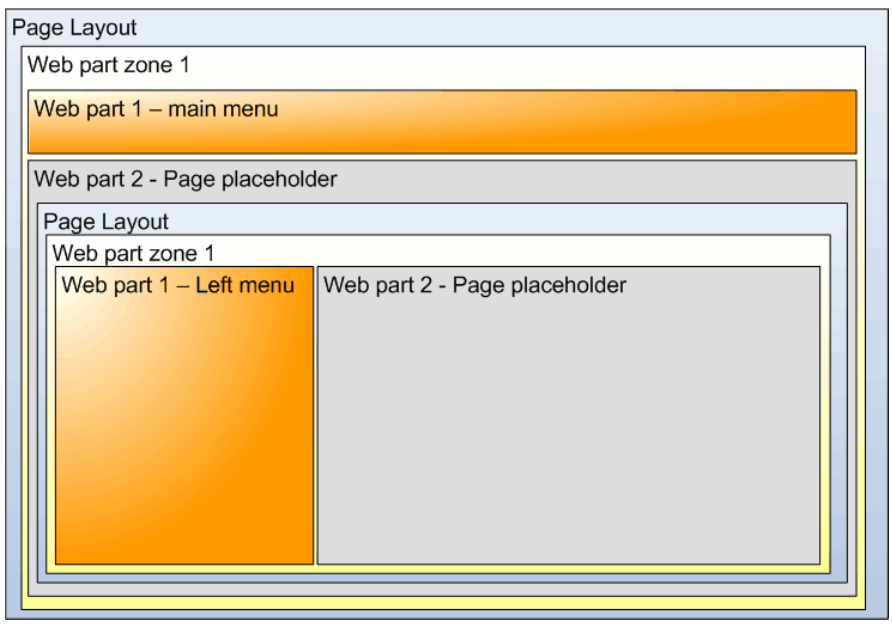
You have learned how to break inheritance of content and how to display content based on the current document type.

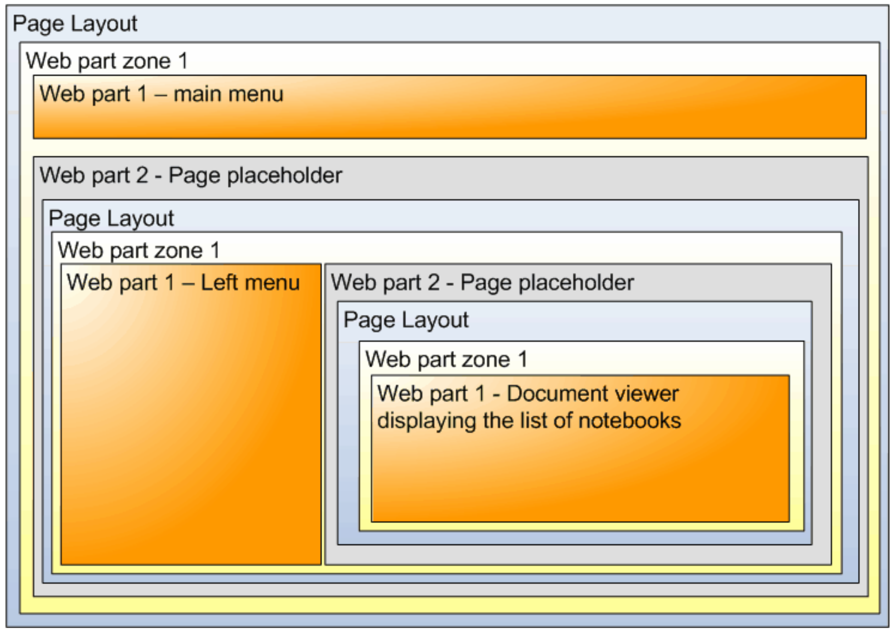
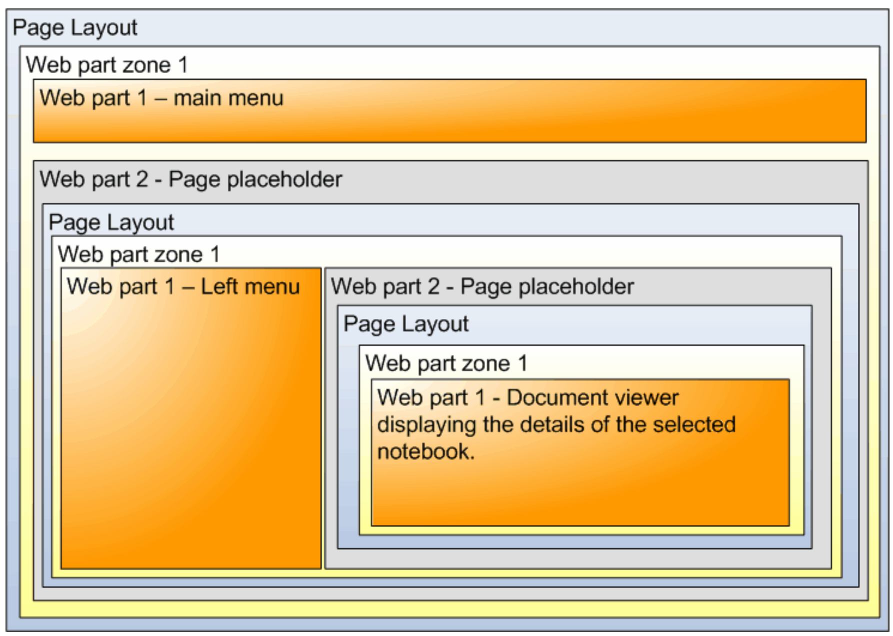
7.2.4.9 Content tree and page templates

The content tree defines not only the site map of your website, but it also how the pages are nested.

For example, when you request the */Products/Laptops/Acer-Aspire-3105WLMi* page, the portal engine loads the content in the following order:

Processed path	Page template	Rendered page

/ (root)	Main menu page template	
/Products	Products page template	

/Products/ Laptops	Laptops page template	
/Products/ Laptops/ Acer-Aspire-3105WLMi	*Inherits page template from parent page. The product detail is displayed automatically instead of the listing by the document viewer.*	

7.2.4.10 Managing page template catalog

You can manage page templates in **Site Manager -> Development -> Page templates**. Here you can organize the templates into categories, configure their properties, create new templates, etc.

These configuration options can also be accessed for the templates of individual pages by selecting the page in the content tree of **CMS Desk**, and using **Edit -> Properties -> Template -> Edit template properties**.

Each template has the following properties:

General tab

On this tab, you can specify the following properties of the page template:

Template display name	The name of the template displayed to the users.
Template code name	The name of the template used in website code.
Category	Here you can choose the category of the page template.
Template description	Description of the template.
Thumbnail	Teaser (preview) image of the template. It is displayed in the **Select template** dialog, e.g. when you are creating a new page.
Template type	Indicates if the page template is a portal page template or ASPX page. If you choose ASPX page you will need to specify the path to its .aspx file using the **Select** button, or by manually entering it, the tilde (~) represents the root directory of your project folder, e.g.: ~/ *CMSTemplates/CorporateSiteASPX/Blog.aspx*
Master template	Indicates if the pages that use the template should be Master pages. Master page templates are usually used for the main menu and logo of a website. Enabling this also causes the template to be selectable as the root master page template in the **New site wizard**.
Inherit content	Configures the visual inheritance as explained in the Visual inheritance topic. You can choose from the following options: • **Inherit all** - inherits the content of all parent pages • **Do not inherit any content** - displays only the current page without any parent content • **Inherit only master page** - inherits from the first master page above the page in the content tree, i.e. if there are more master pages, it inherits only from the one which is the closest above it in the hierarchy • **Select inherited levels** - inherits only content from the chosen content tree levels

Sites tab

On this tab, you can choose which websites the page template will be available for.

Scopes tab

Please refer to the Page template scopes topic for information about this tab.

Layout tab

Here you can edit the HTML layout of the page template. You can either choose to use some of the pre-defined ("shared") layouts or you can specify your own HTML layout code ("custom layout"). Learn more in the [Page layouts] topic.

Web parts

Here you can see the XML document with the configuration of the web parts. Use this dialog only in a situation when the standard **Design** mode isn't working as expected (e.g. due to some data inconsistency).

Header

Here you can specify custom HTML code that will be added to the **<head>** element (it will be placed between the `<head></head>` tags) of all pages that use this page template. This is useful if you need to link some additional CSS or JavaScript files.

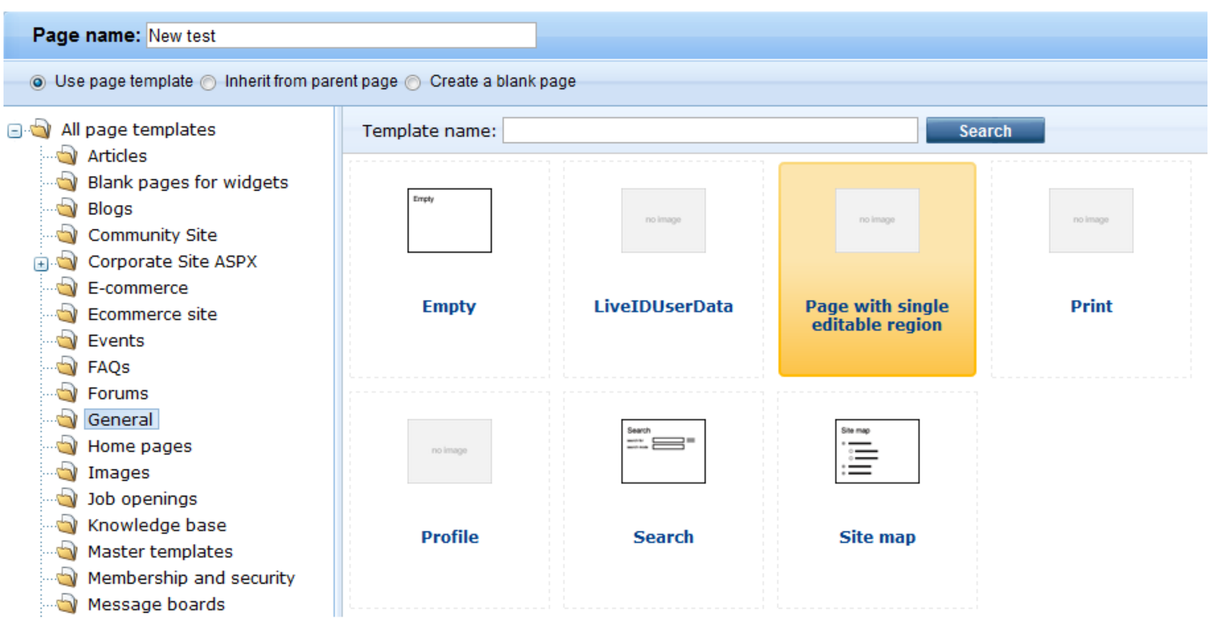
Documents tab

On this tab, you can see a list of documents (pages) that are using the page template. By clicking a document in the list, you will be redirected to the document's **CMS Desk -> Edit** mode. The documents in the list can be filtered according to the **Site** they are on, their **Document name** and their **Document type**.

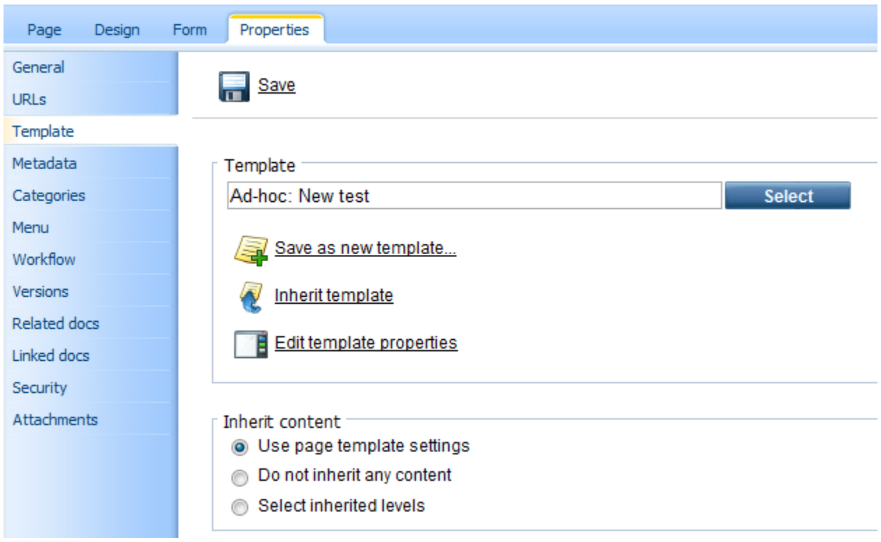
7.2.4.11 Cloning and modifying a page template

You may need to create a new page that will be similar to existing pages, but with some minor modification of the page template. Imagine you want to display the editable region from the previously created page template in a container box.

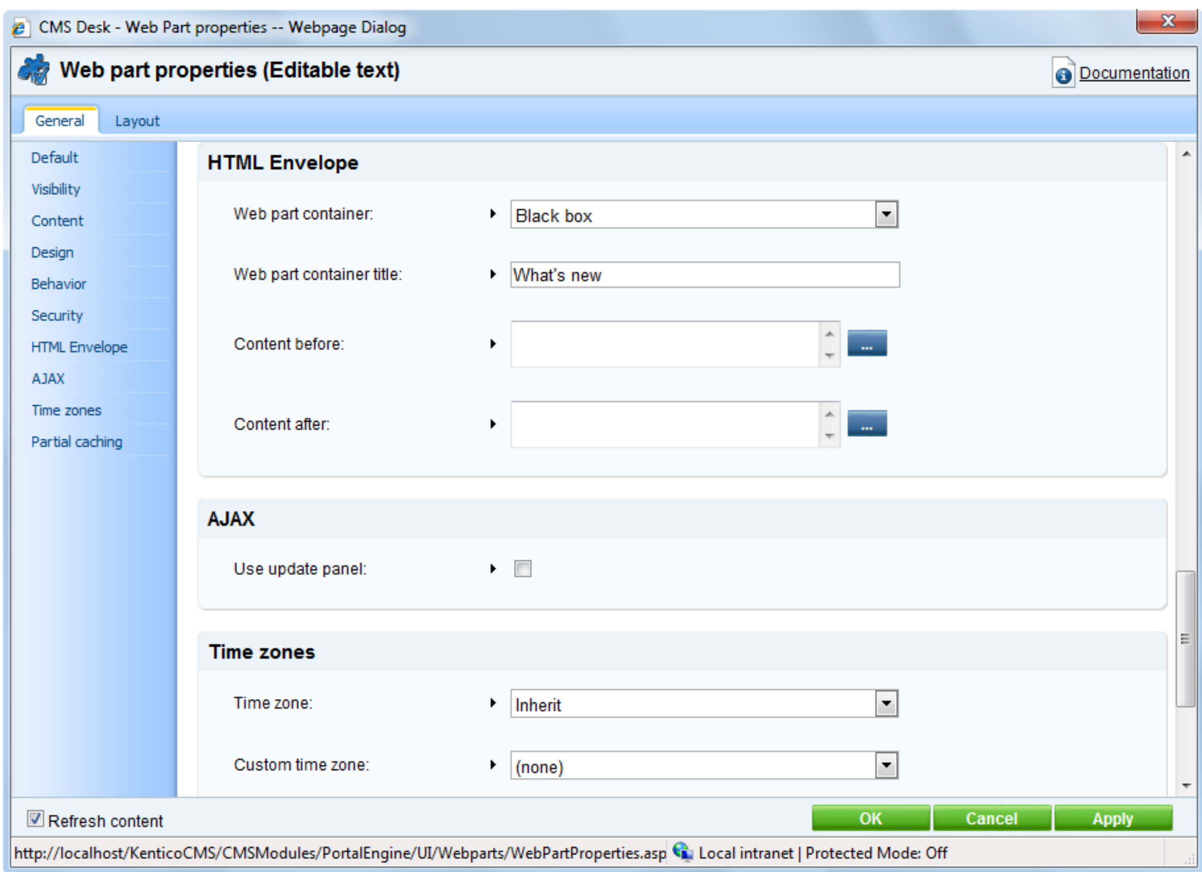
Instead of creating the page template from scratch, simply create a new page based on your existing page template.



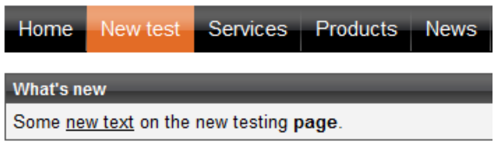
Then choose the page, click **Properties -> Template**, click **Clone as ad-hoc template** and click **Save**. A new ad-hoc template is now created and you can edit it without modifying the existing pages based on the original template:



Now switch to the **Design** tab and click the **Configure** ( ) button of the **Editable text** web part. Select the *Black box* container from the **Web part container** drop-down list and enter *What's new* into the **Web part container title** field:



Click **OK**. Switch to the **Page** tab and enter some text into the editable region. Click **Save** and click **Live site** in the main toolbar. You will see a page like this:



As you can see, the text is now surrounded by a black container. When you check the original **My test** page, you will see it uses the original design and it wasn't affected by the change we made to our new copy of the page template.

Re-using the adjusted template again

If you want to re-use the new, adjusted page template, you can click the **Save as new template** button in the **Properties -> Template** dialog as you did previously.

Important!

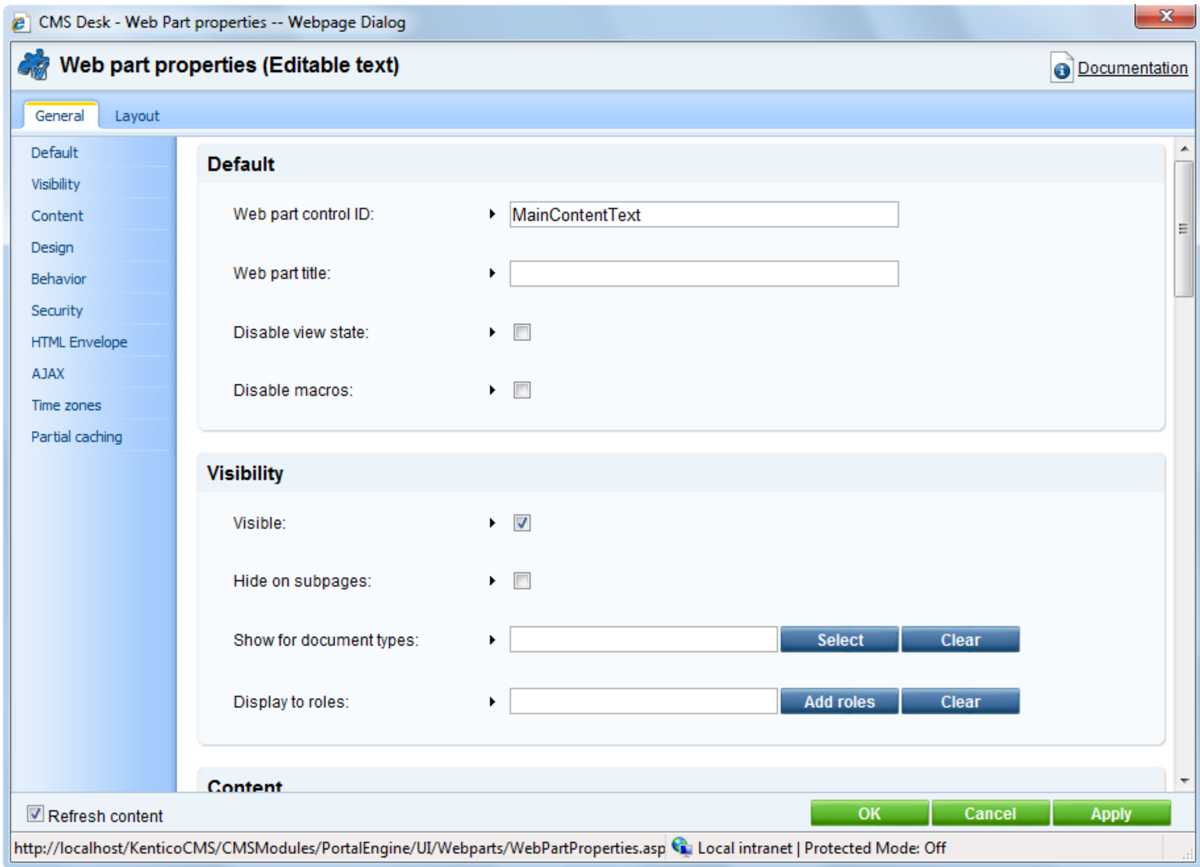
Please keep in mind that when you create a page based on an ad-hoc page template and later delete the page, the corresponding ad-hoc page template is deleted as well and **cannot be restored**.

7.2.4.12 Using and configuring web parts

Web parts can be used on both portal page templates and ASPX page templates. However, with ASPX page templates, you lose the friendly user interface and need to set the properties in Visual Studio.

Go to **CMS Desk -> Content**, click some page and choose **Design** in the toolbar. You will see the page template structure like this:

Every web part can be moved up/down, to another web part zone and it can be configured using the **Configure** (⚙) button. It opens the **Web part properties** dialog like this:



When you click **OK**, the changes are saved and the window closes. When you click **Apply**, the changes are saved, but you can continue modifying the properties. All changes are applied immediately, the page templates are not connected with workflow or versioning.

Impacts of changing page templates

When you change a re-usable page template that is used by several pages, the changes will appear on all page templates.

If you need to modify only a single page structure, you need to clone the page template as an ad-hoc template or as a new page template. See also Cloning and modifying a page template.

7.2.4.13 Web part binding (obsolete)

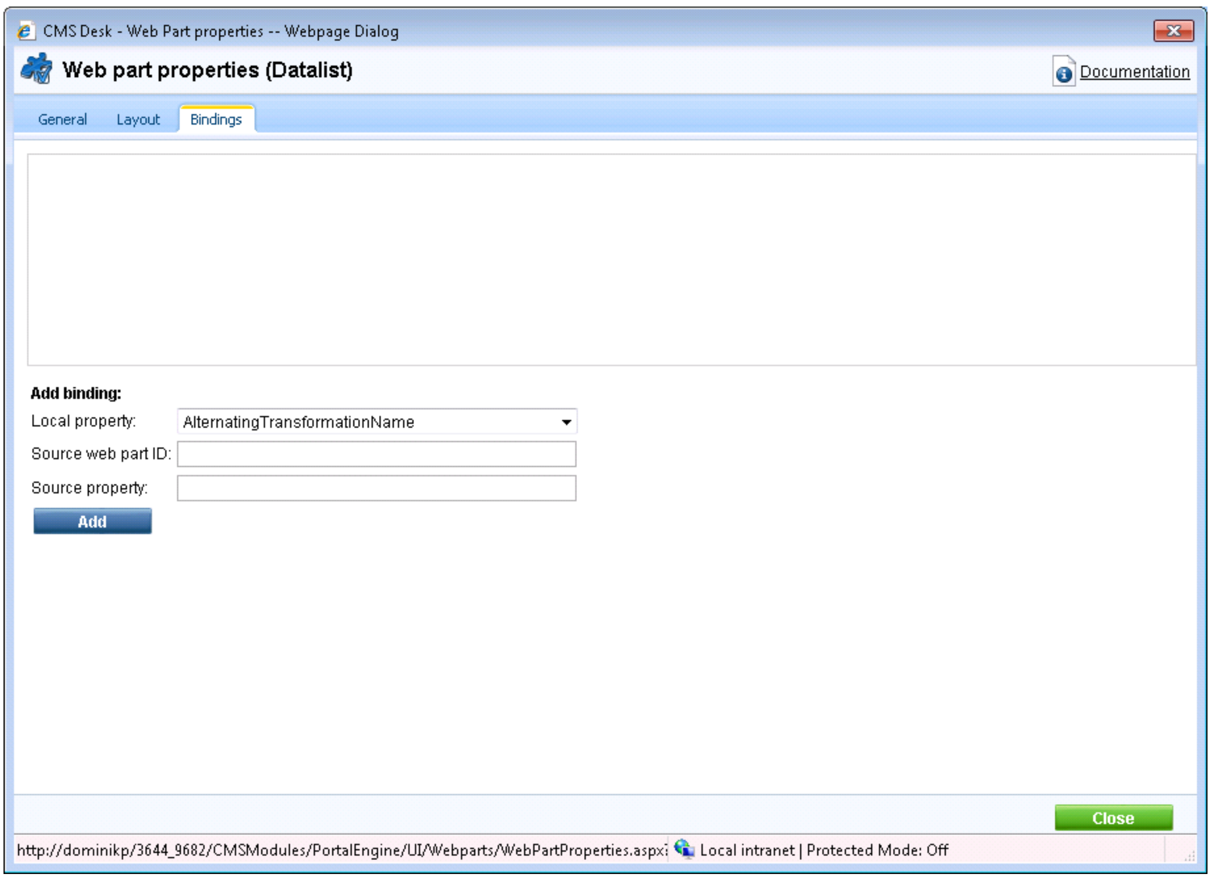
Obsolete feature

This feature is now obsolete. If you need to bind the behavior of some web part to another web part value, please use URL parameters or create a user control that contains both web parts and write custom code that will ensure the communication between the web parts.

If you need to use this feature for backward compatibility, you need to add the following parameter to your web.config file, to the **appSettings** section:

```
<add key="CMSShowWebPartBindingTab" value="true"/>
```

Web part binding allows you to connect two web parts. For example: you can have a web part containing a drop-down list with countries. When some value is selected, it is provided to another web part that displays a list of company offices in the selected country. You can manage web part binding on the **Binding** tab:



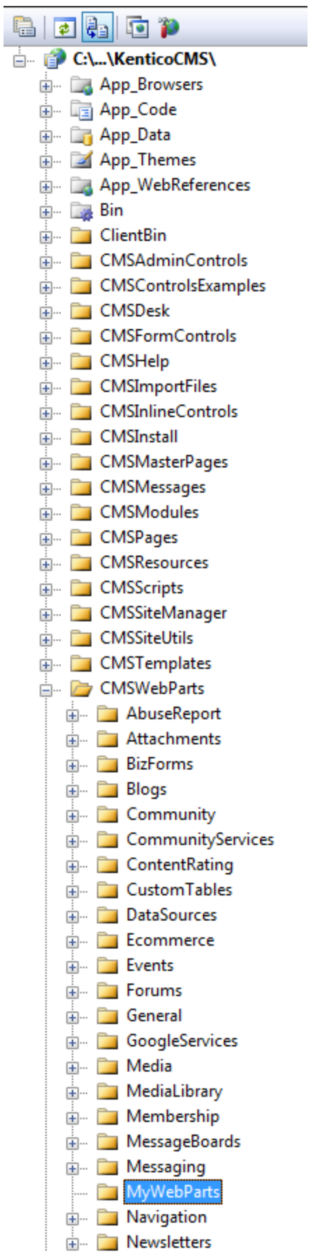
Alternative option

If binding doesn't work as expected, you can alternatively create your own user control, place the required web parts on it and manage their interactions using your own code. You can find more details on creating your own user control in the Adding custom code to the page topic.

Example - creating a product selection dialog with drop-down list and product list

In this example, we will create two web parts - selector and viewer and bind their properties so that the product viewer reflects the product selector status.

1. Open the CMS project in Visual Studio and create a new folder MyWebParts under CMSWebParts folder in the Solution Explorer:



2. Create a new user control (ASCX) under MyWebParts folder and call it `ProductSelector.ascx`. Switch to its HTML source and paste the following code:

```
<asp:DropDownList ID="DropDownList1" runat="server"
    OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged" AutoPostBack
="true">
    <asp:ListItem Value="Under">Under 250</asp:ListItem>
    <asp:ListItem Value="Over">Over 250</asp:ListItem>
</asp:DropDownList>
```

Switch to the code behind and add the following code to the beginning of the code:

[C#]

```
using CMS.PortalControls;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.CMSHelper;
using CMS.ExtendedControls;
```

Change the class inheritance like this:

[C#]

```
public partial class CMSWebParts_MyWebParts_ProductSelector : CMSAbstractWebPart
```

Add the following code inside the page class:

[C#]

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    this.SetValue("PriceRange", this.DropDownList1.SelectedValue.ToString());
    this.ReloadConsumers();
}
```

Please note that you need to set the new value of the web part property `PriceRange` and call the `ReloadConsumers` method.

3. Create a new user control (ASCX) under MyWebParts folder and call it `ProductViewer.ascx`. Switch to its HTML source and paste the following code:

```
<%@ Register Assembly="CMS.Controls" Namespace="CMS.Controls" TagPrefix="cc1" %>
<cc1:CMSRepeater ID="CMSRepeater1" runat="server" EnableViewState="true"
     TransformationName="cms.product.default" SelectedItemTransformationName="cms.
product.default"
     Path="/%" ClassNames="cms.CellPhone">
</cc1:CMSRepeater>
```

The CMSRepeater web part displays the product list.

Switch to the code behind and add the following code to the beginning of the code:

[C#]

```
using CMS.PortalControls;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.CMSHelper;
using CMS.ExtendedControls;
using CMS.PortalControls;
```

Change the class inheritance like this:

[C#]

```
public partial class CMSWebParts_MyWebParts_ProductViewer : CMSAbstractWebPart
```

Add the following code inside the page class:

[C#]

```
/// <summary>
/// Content loaded event handler
/// </summary>
public override void OnContentLoaded()
{
    base.OnContentLoaded();
    SetupControl();
}

/// <summary>
/// Reloads data on request
/// </summary>
public override void ReloadData()
{
    base.ReloadData();
    this.SetupControl();
    this.CMSRepeater1.ReloadData(true);
}

/// <summary>
/// Initializes the control properties
/// </summary>
protected void SetupControl()
{
    if ((this.PagePlaceholder.ViewMode == ViewModeEnum.Design) || (this.
HideOnCurrentPage) || (!this.IsVisible))
    {
        // Stop processing in Design mode and if the control is invisible
        this.CMSRepeater1.StopProcessing = true;
    }
    else
    {
        // set CMSRepeater properties according to the selected value (price
range)
        string priceRange = ValidationHelper.GetString(this.GetValue("PriceRange"
), "Under");
        if (priceRange == "Over")
        {
            this.CMSRepeater1.WhereCondition = "SKUPrice > 250";
        }
        else
        {
            this.CMSRepeater1.WhereCondition = "SKUPrice <= 250";
        }
    }
}
```

4. Save the changes and go to **Site Manager -> Development -> Web parts**. Create a new category called *My Web Parts* and add a new web part:

- **Web part display name**: Product selector
- **Web part code name**: ProductSelector
- **Web part file name**: MyWebParts/ProductSelector.ascx

5. Create another new web part:

- **Web part display name**: Product viewer

- **Web part code name**: ProductViewer
- **Web part file name**: MyWebParts/ProductViewer.ascx

... and then switch to the **Properties** tab and create a new property:

- **Attribute name**: PriceRange
- **Attribute type**: Text
- **Attribute size**: 100
- **Allow empty value**: yes
- **Display attribute in the editing form**: no

6. Go to CMS Desk and create a new blank page using the Simple page layout and call it **Product list**.

7. Switch to the **Design** tab and add the web parts **Product selector** and **Product viewer** on the page.

8. Configure the **Product viewer** web part and choose the **Binding** tab. Add a new web part binding using the following values:

- **Local property**: PriceRange
- **Source web part ID**: ProductSelector
- **Source property**: PriceRange

9. Go to the Live site mode and see the page. It should look like this:

When you change the drop-down list value, the product list below is automatically updated.

7.2.4.14 Adding custom code to web parts (obsolete)

Obsolete feature

This feature is now obsolete. If you need to customize the behavior of some web part, please clone the web part and customize it.

You can find more details in the Modifying web part behavior topic.

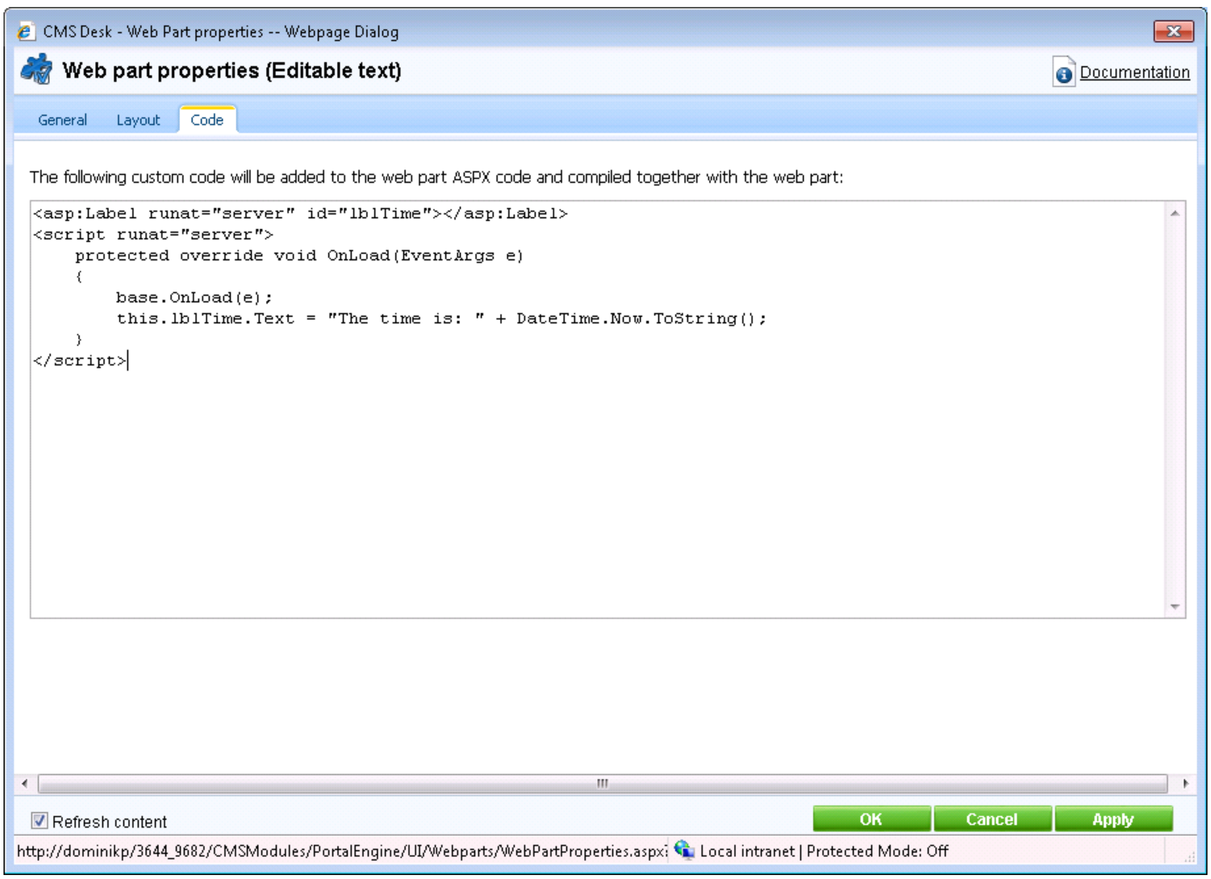
If you still need to use this feature (for backward compatibility), you need to enable it by adding the following parameter to your web.config file, to the **appSettings** section:

```
<add key="CMSShowWebPartCodeTab" value="true" />
```

Limitations when you use the Code tab

If you add custom code on the Code tab, you will not be able to pre-compile the website and you will not be able to use the website in medium trust environment. If this is an issue for you, you can use an alternative solution described in the [Modifying web part behavior] topic.

If you need to modify the behavior or enhance the functionality of some web part only on a single page template, you can add custom code on the **Code** tab:



The following code displays the current date and time below the current web part:

[C#]

```
<asp:Label runat="server" id="lblTime"></asp:Label>
<script runat="server">
    protected override void OnLoad(EventArgs e)
    {
        base.OnLoad(e);
        this.lblTime.Text = "The time is: " + DateTime.Now.ToString();
    }
</script>
```

Programming language

Please note: You can only use the programming language in which the web part is written. If the web part was written in C#, you can use only C# here.

The following code sample sets the **WhereCondition** property of the web part dynamically at run-time:

[C#]

```
<script runat="server">
public override void OnContentLoaded()
{
this.SetValue("WhereCondition", "NewsTitle LIKE '%News%'");
base.OnContentLoaded();
}
</script>
```

Calling the original method from the inherited class

If you override some method of the web part, please be sure to always also call the original method (using base.Method in C# or MyBase.Method in VB). If you omit this, the web part may not work properly.

7.2.4.15 Common web part properties

Many web parts use the same or similar properties. The following table summarizes the most important properties found in most web parts:

Default properties

Property Name	Description	Sample Value
Web part control ID	Unique identifier of the web part.	text1

Visibility

Property Name	Description	Sample Value
Visible	Indicates if the web part is visible.	
Hide on sub-pages	Indicates if the web part should be hidden on sub-pages. If you check this box, the web part will be displayed only on the page that uses the page template with this web part and it will be hidden on sub-pages that inherit the page template.	
Show for document types	Indicates for which document types the web part should be displayed. If the type of the currently selected document is different, the web part will not be displayed.	cms.news;cms.event
Display to roles	List of roles whose members should see the web part. If no role is specified, the web part zone is displayed to all users.	cmseditors;customrole

Web part container

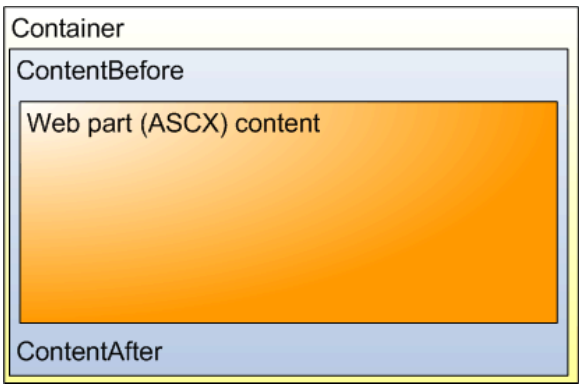
Property Name	Description	Sample Value
Web part container	Container (box) displayed around the web part. The containers can be defined in **Site Manager -> Development -> Web part containers**.	
Container title	Container title displayed in the web part container if one is specified. Displayed only if the *{%ContainerTitle%}* macro is used in the code of the container.	Latest news
Container CSS class	CSS class used for the web part container. Applied only if the *{%ContainerCSSClass%}* macro is used as a value of the *Class* attribute in the code of the container.	
Container custom content	Custom content of the web part container; applied only if the *{%ContainerCustomContent%}* macro is used in the code of the container.	

HTML Envelope

Property Name	Description	Sample Value
Content before	HTML content to be displayed before the web part. Can be used to display a header or	<table style="background-color: red"><tr><td>

	add some encapsulating code such as <*div*> or <*table*> elements for the required layout.	
Content after	HTML content to be displayed 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.	</td></tr></table>

The structure of a web part and its envelope looks like this:



You can find more details on web part containers in the Containers overview topic.

Content

Property Name	Description	Sample Value
Path	Path of the documents to be displayed. See Appendix B - Path expressions for details.	/news/%
Highlighted node path	Alias path of the node that should be selected in a menu control. If you do not specify any value, the current path is used.	/products/nokia
Query	Name of the query to be used for retrieving data from the Kentico CMS Database. The queries can be defined in **Site Manager -> Document types -> edit some document type -> Queries**.	cms.news.selectlatest

Content filter

Property Name	Description	Sample Value
Document types	Types of documents that should be displayed, separated by a semicolon (;). In the case of menu and navigation web	cms.article;cms.news;cms.menuitem

	parts, page (cms.menuitem) documents are selected by default. The other web parts retrieve all documents without their custom fields by default. Please note: if you specify more than one document type, only common data fields from the *View_CMS_Tree_Joined* view will be available in the web part (in transformations, WHERE condition and ORDER BY expressions).	
Combine with default culture	Indicates if the default language version of the document should be displayed if the document is not translated to the current language.	You can choose between **Yes** and **No** or you can use website-level settings.
Culture code	Culture of the content to be displayed.	en-us
Maximum nesting level	Maximum nesting level. It specifies the number of sub-levels in the content tree that should be included in the displayed content. The value 1 indicates that only the current document should be returned. The value -1 indicates all child documents.	
ORDER BY expression	ORDER BY part of the SELECT query.	ProductName ASC, ProductPrice DESC
Select only published	Indicates if only published documents should be displayed.	
Site name	Code name of the website from which you want to display the content. If you leave the value empty, the content is retrieved from the current website.	CorporateSite
WHERE condition	WHERE part of the SELECT query.	ProductPrice > 100 AND ProductColor='green'
Filter out duplicate documents	If the displayed data contains multiple links to the same document (see Linked docs for details), you can choose to display only one of them.	

System settings

Property Name	Description	Sample Value
Check permissions	Indicates if the permissions of the current user should be checked for the content	

	displayed by the web part. If enabled, only documents for which the user has the *read* permission will be displayed.	
Cache item name	Sets the name of the cache key used for the content displayed by the web part. If not specified, this name is generated automatically based on the site, document path, **Web part control ID** and current user. A cache key can be shared between multiple web parts displaying the same content on different pages in order to avoid keeping redundant data in the memory.	latestnewslist
Cache minutes	Sets the number of minutes for which the content displayed by the web part should remain cached before its latest version is reloaded from the database. If left empty, the value entered into the **Site Manager -> Web site -> Cache content (minutes)** setting will be used instead. If set to *0*, caching will be disabled for the web part.	10
Cache dependencies	Contains a list of cache keys on which the content cache of the web part depends. When the specified cache items change, the content 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 content of the specific web part.	

Design

Property Name	Description	Sample Value
CSS prefix	Prefix used for CSS class names. This property allows you to set up different CSS styles for particular menu levels. Every level of the menu will use the prefix for CSS class names that you specify.	Main;Sub1;Sub2
Highlight all items in path	Indicates if all items in the currently selected path of the menu control should be displayed	

	as highlighted.	
Submenu indicator	Path to the image that should be displayed next to items that have sub-items.	/images/submenu.gif
Use alternating styles	Indicates if odd and even items should have different styles.	

Paging

Property Name	Description	Sample Value
Enable paging	Indicates if displayed data should be paged.	
Paging mode	Type of paging parameter - it can be passed either through the URL (**Query string**) or through postback (**Postback**).	
Pager position	Position of the pager - top or bottom	
Page size	Number of records per page.	10
Query string key	The name of the URL parameter that will contain the current page number.	mylistpage
Show first and last buttons	Indicates the buttons that link to the first and last page should be displayed.	

Relationships

These settings allow you to configure the web part so that it displays only content that is related to the specified (main) document.

Property Name	Description	Sample Value
Main document	Document for which you want to display related documents.	
Relationship name	Name of the relationship between documents.	Is related to
Main document is on the left side	Indicates if the main document is on the left side of the relationship.	

No data behavior

Property Name	Description	Sample Value
Hide if no record found	Indicates if the web part should be hidden when no record is found.	
No record found text	Text that should be displayed if no data is	No data found.

	found.	

Editing buttons

Property Name	Description	Sample Value
Show New button	Indicates if the button for adding new items should be displayed in the **Edit** mode when viewing the page.	
New button text	New button description text.	Add new news.
Show edit and delete buttons	Indicates if edit and delete buttons should be automatically shown for each displayed item in the **Edit** mode.	

Transformations

Property Name	Description	Sample Value
Alternating transformation	Transformation used for even items in list view mode in format *<class prefix>.<document type>.<transformation name>*.	cms.news. preview_alternating
Transformation	Transformation used for items in list view mode in format *<class prefix>.<document type>.<transformation name>*.	cms.news.preview
Selected item transformation	Transformation used for items in detail view mode in format *<class prefix>.<document type>.<transformation name>*.	cms.news.default
Item separator	Item separator displayed between records.	

7.2.4.16 Path and macro expressions in web part properties

Many web parts use the **Path** property to specify what content from the content tree should be displayed. You can find all path format options in Appendix B - Path expressions.

Also, all web part properties support macro expressions that allow you to insert a dynamic value instead of a constant in the property. Such a dynamic value is evaluated at run-time and allows you to specify context-dependent values. See Appendix A - Macro expressions for details.

7.2.4.17 Adding custom code to the portal page template

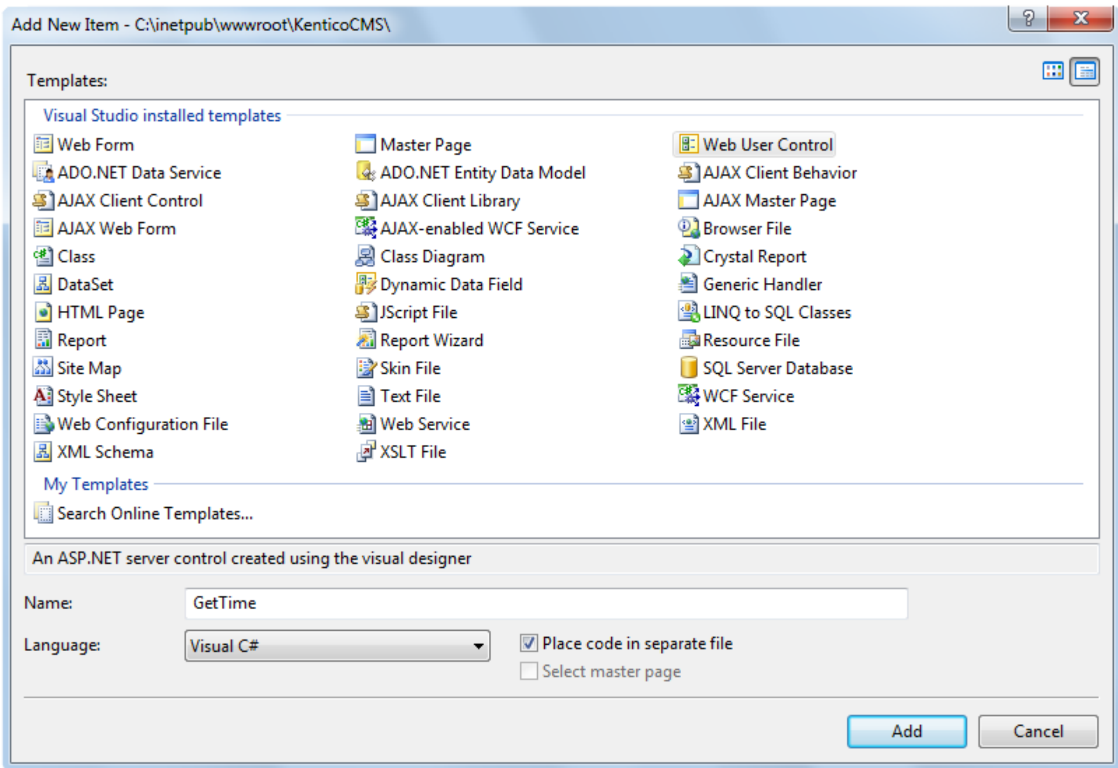
The easiest way to insert custom code into a portal engine-based website is using standard ASCX user controls. This chapter will show you how to do this. If you're not familiar with Visual Studio development, you can skip this chapter.

Current time example

In this example, we will create a simple user control (ASCX) using Visual Studio 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 in the Solution Explorer window and click **New Folder**. Name the folder the same as the code name of your site, e.g. CorporateSite - this folder will be exported with your project when you decide to export the site and import it on a live server.

Right-click the new folder and click the **Add new item...** option. Choose to create a new Web User Control and set its name to **GetTime.ascx**. You can set the programming language option to either Visual C# or Visual Basic.



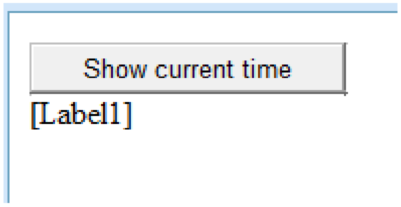
Click **Add**. Switch to the **Design** tab and drag and drop the following controls and set their properties:

Button control:

- **ID**: Button1
- **Text**: Show current time

Label control:

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



Double-click the **Show current time** button and enter the following code into 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 - the user controls are compiled at run time.

Save all changes.

Adding the user control on the page

Sign in to **Kentico CMS Desk**, click the **Home** page and click **Design**. Remove the web part in the **zoneCenter** zone and click the **Add web part** ( ) button in this zone. Choose the **General/User control** web part. Enter the following value in the **User control virtual path** property:

~/CorporateSite/GetTime.ascx

(the folder name must reflect the folder that you previously created)

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

In this short example, you have seen that you can easily add any custom code developed as an ASCX user control in Visual Studio. This user control can contain any .NET controls, third-party controls or ADO.NET code that will retrieve data from an external database.

User controls versus web parts

Another option how to insert custom code into the page is creating your own web part. A web part is basically an ASCX user control, but it inherits some standardized properties and methods from the *CMSAbstractWebPart* class. You will usually build web parts in case you need to create re-usable, parameterized user controls. Web part development is described in the Development -> Web parts -> Developing web parts topic.

7.2.4.18 Displaying data from an external database or Web Service

Besides displaying Kentico CMS content, you can also display data from your external database or Web Service. In this case, you need to develop a user control (ASCX) that will use ADO.NET to retrieve the data or that will contact the Web Service and call its methods. Since you can place any custom code into the user control, you will simply use the standard ASP.NET code you would use when creating the website from scratch.

Example: Retrieving data from the sample Northwind database

In this simple example, you will see how to display data on the sample Corporate Site from the *Categories* table of the Northwind database using ADO.NET. You may need to use some other database if you do not have the sample Northwind database on your server.

Open the web project in Visual Studio using the **WebProject.sln** file. Create the folder **CorporateSite** and create a new user control **CustomData.ascx** in this folder. It's important to create the control in this folder so that it's exported with your website later, when you decide to import the website on your live server.

Drag and drop the standard ASP.NET **GridView** control on your user control and set its ID to **GridView1**.

Add the following line to the beginning of the code behind file:

[C#]

```
using System.Data.SqlClient;
```

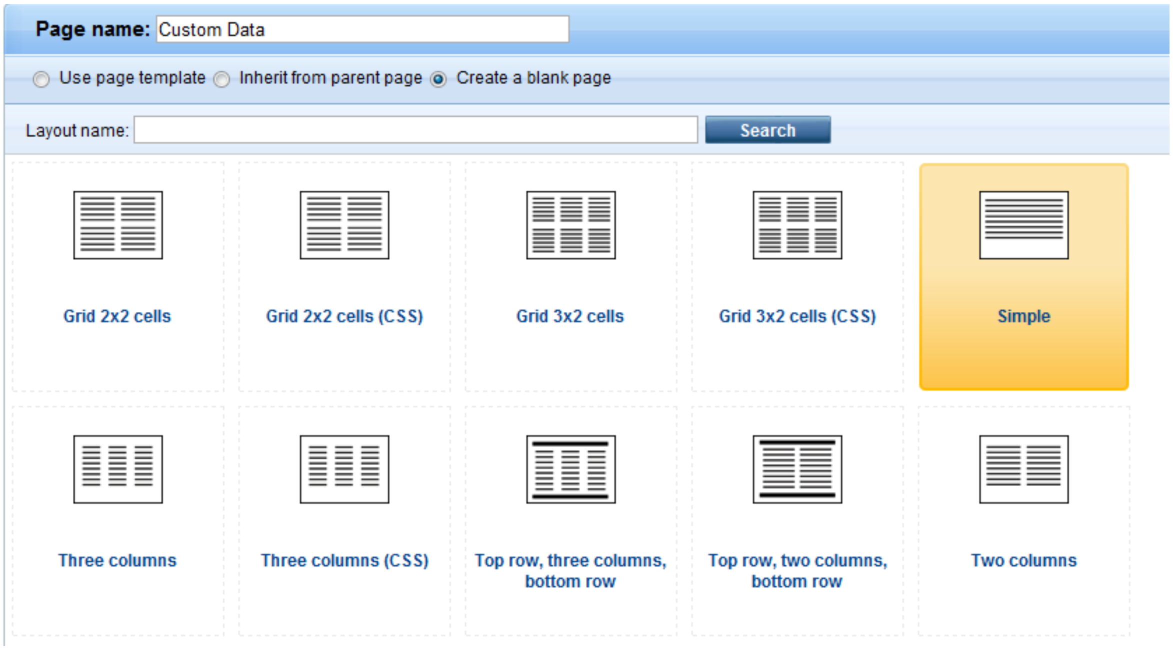
Add the following code to the **Page_Load** method:

[C#]

```
// create sql connection - you could use Oracle or OLEDB provider as well

SqlConnection cn = new SqlConnection("Persist Security Info=False;
database=northwind;server=server1;user id=sa;password=psswd;Current
Language=English;Connection Timeout=120;");

// create data adapter
SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM categories", cn);
DataSet ds = new DataSet();

// fill the dataset with data from database
da.Fill(ds);

// bind data to the grid view
GridView1.DataSource = ds;
GridView1.DataBind();
```

Save all changes.

Sign in as administrator to **CMS Desk**, select the **/Examples** document in the content tree and create a new **Page (menu item)**. Call it **Custom Data** and choose the **Create a blank page** option. Select the **Simple** layout and click **Save**.



Switch to the **Design** tab, add a new web part **General/User control** and set its **User control virtual path** property value to *~/corporatesite/customdata.ascx*.

Click **OK** and switch to the **Live site** mode of the page. You will see a grid with data from the external database:

CategoryID	CategoryName	Description
1	Animals	Dogs, cats, horses
2	Colors	Red, green, blue, orange,...
3	Names	John, Kevin, Julia, Robin

As you can see we used standard ASP.NET methods to display external data on the website.

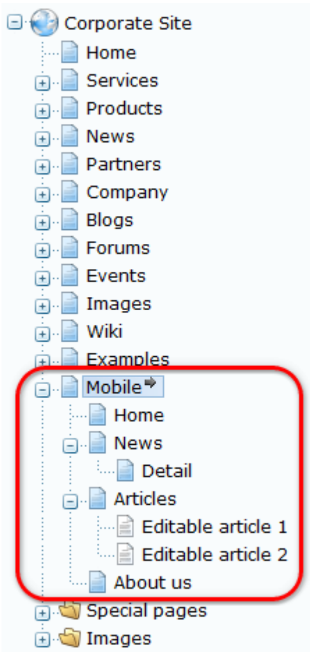
7.2.4.19 Developing sites for mobile devices

Kentico CMS allows you to create a dedicated site section where users accessing the site from mobile devices will be redirected.

An example of such a site section can be found on the sample **Corporate Site**, under the **/Mobile** node. The **Mobile** page works as a master page for the mobile section. It has no template inheritance configured so that it doesn't inherit from the site's main master page. Under it, you can find the following sections:

- **Home** – this is the page where the mobile user gets redirected from the site's main *Home* page; the content of the page is loaded by the *Repeater* web part dynamically from the *Editable text* web part on the main *Home* page
- **News** – this page contains a *Repeater* web part displaying the *News* documents stored under */News*; its transformation is written so that when you click a news item's title, you get redirected to the */Mobile/News/Detail* page with the *ID* of the news item in the wildcard part of the URL (*/Mobile/News/{id}*)

- **Articles** – the *Repeater* web part on this page displays the articles stored under it; the content of this section is not shared with the rest of the site and is used only in the mobile section
- **About us** – this page contains only two *Editable text* web parts and its content is separate and used only in the mobile section



Mobile device recognition

Mobile device recognition is ensured by the **Mobile device redirection** web part placed on the **Home** page. Site visitors accessing the site's main **Home** page from a mobile device get recognized by the web part and are redirected to the URL specified by its **Redirection URL** property. In this case, they get redirected to the mobile section of the website, i.e. *~/Mobile/Home.aspx*.

The figures below show how the page looks when viewed on different mobile devices:

CSS stylesheet for the mobile section

There is a dedicated CSS stylesheet - **Corporate Site – Mobile** - which is used only by the pages in the mobile section. While the site's main stylesheet is **Corporate Site**, the pages in the mobile section have the **Corporate Site – Mobile** stylesheet assigned in **Properties -> General -> CSS stylesheet**.

7.2.4.20 Portal engine development internals and API

7.2.4.20.1 Database tables and API classes

The following database tables are used during portal engine development:

- **CMS_Layout** - contains pre-defined page layouts and information about them
- **CMS_PageTemplate** - contains page templates and their settings
- **CMS_PageTemplateCategory** - contains page template categories and information about them
- **CMS_PageTemplateScope** - contains page template scopes (used to limit where a page template can be used)
- **CMS_PageTemplateSite** - assigns page templates to the sites where they can be used

The following **CMS.PortalEngine** namespace classes can be used to perform certain portal engine development tasks through the API:

- **PageTemplateInfo**, **PageTemplateInfoProvider** - these classes provide functionality for managing page templates
- **PageTemplateCategoryInfo**, **PageTemplateCategoryInfoProvider** - these classes provide functionality for managing page template categories
- **PageTemplateScopeInfo**, **PageTemplateScopeInfoProvider** - these classes provide functionality for managing page template scopes

As well as the following **CMS.SiteProvider** namespace classes:

- **LayoutInfo**, **LayoutInfoProvider** - these classes provide functionality for managing page layouts

The following topics show examples of how these classes can be used:

- **Page template examples:**
  - Getting page template data
  - Managing page templates and their categories
  - Assigning and removing page templates from sites
  - Managing page template scopes
- **Page layout examples:**
  - Getting page layout data
  - Managing page layouts

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

7.2.4.20.2 Page templates

7.2.4.20.2.1 Getting page template data

The following sample code shows how you can get a page template as a *PageTemplateInfo* object:

[C#]

```
using CMS.PortalEngine;

...

        // Get page template object by template code name
        PageTemplateInfo pti = PageTemplateInfoProvider.GetPageTemplateInfo("cms.art

        // Get page template object by template ID
        PageTemplateInfo pti2 = PageTemplateInfoProvider.GetPageTemplateInfo(10);
```

The following sample code shows how you can get a DataSet containing page templates from the system:

[C#]

```
using System.Data;
using CMS.PortalEngine;
```

```
...

        string where = "PageTemplateID > 10";
        string orderby = "PageTemplateDisplayName";

        // Get DataSet of page templates according to the given parameters
        DataSet ds = PageTemplateInfoProvider.GetTemplates(where, orderby);
```

7.2.4.20.2.2 Managing page templates and their categories

The following sample code shows how you can create a new page template category under the root category:

[C#]

```
using CMS.PortalEngine;

...

        // Get parent for new category
        PageTemplateCategoryInfo parent = PageTemplateCategoryInfoProvider.GetPageTe

        // If parent exists
        if (parent != null)
        {
            // Create new page template category object
            PageTemplateCategoryInfo ptci = new PageTemplateCategoryInfo();

            // Set properties
            ptci.DisplayName = "Test category";
            ptci.CategoryName = "TestCategory";
            ptci.ParentId = parent.CategoryId;

            // Save to database
            PageTemplateCategoryInfoProvider.SetPageTemplateCategoryInfo(ptci);
        }
```

The following sample code shows how you can create a new page template under an existing category and add it to the system:

[C#]

```
using CMS.PortalEngine;
using CMS.CMSHelper;

...

        // Get parent category
        PageTemplateCategoryInfo ptci = PageTemplateCategoryInfoProvider.
```

```
GetPageTemplateCategoryInfo("TestCategory");

        // If parent exists
        if (ptci != null)
        {
            // Create new page template object
            PageTemplateInfo pti = new PageTemplateInfo();

            // Set properties
            pti.DisplayName = "Test page template";
            pti.CodeName = "TestPageTemplate";
            pti.Description = "This is page template created by API Example";
            pti.PageTemplateSiteID = CMSContext.CurrentSiteID;
            pti.FileName = " ";
            pti.ShowAsMasterTemplate = false;
            pti.IsPortal = true;
            pti.InheritPageLevels = ""; // inherits all
            pti.IsReusable = true;

            // Template layout with a web part zone
            pti.PageTemplateLayout = "<cc1:CMSWebPartZone ID=\"zoneLeft\" runat=\"se

            pti.CategoryID = ptci.CategoryId;

            // Save to database
            PageTemplateInfoProvider.SetPageTemplateInfo(pti);
        }
```

The following sample code shows how you can modify an existing page template:

[C#]

```
using CMS.PortalEngine;

...

        // Get template object
        PageTemplateInfo pti = PageTemplateInfoProvider.GetPageTemplateInfo("TestPag

        // If template exists
        if (pti != null)
        {
            // Update properties
            pti.DisplayName += " updated";
            pti.PageTemplateLayout += " updated";

            // Update data in database
            PageTemplateInfoProvider.SetPageTemplateInfo(pti);
        }
```

The following sample code shows how you can delete an existing page template:

[C#]

```
using CMS.PortalEngine;

...

        // Get template object
        PageTemplateInfo pti = PageTemplateInfoProvider.GetPageTemplateInfo("TestPag

        // If template exists
        if (pti != null)
        {
            // Delete
            PageTemplateInfoProvider.DeletePageTemplate(pti);
        }
```

The following sample code shows how you can delete an existing page template category:

[C#]

```
using CMS.PortalEngine;

...

        // Get template category object
        PageTemplateCategoryInfo pti = PageTemplateCategoryInfoProvider.
GetPageTemplateCategoryInfo("TestCategory");

        // If category exists
        if (pti != null)
        {
            // Delete
            PageTemplateCategoryInfoProvider.DeletePageTemplateCategory(pti);
        }
```

7.2.4.20.2.3 Assigning and removing page templates from sites

The following sample code shows how you can assign a page template to the current site:

[C#]

```
using CMS.PortalEngine;
using CMS.SiteProvider;

...

        // Get site and page template objects
        PageTemplateInfo pti = PageTemplateInfoProvider.GetPageTemplateInfo("TestPag
        SiteInfo si = SiteInfoProvider.GetCurrentSite();
```

```
        // If objects exist
        if ((pti != null) && (si != null))
        {
            PageTemplateInfoProvider.AddPageTemplateToSite(pti.PageTemplateId, si.Si
        }
```

The following sample code shows how you can remove a page template from the current site:

[C#]

```
using CMS.PortalEngine;
using CMS.SiteProvider;

...

        // Get site and page template objects
        PageTemplateInfo pti = PageTemplateInfoProvider.GetPageTemplateInfo("TestPag
        SiteInfo si = SiteInfoProvider.GetCurrentSite();

        // If objects exist
        if ((pti != null) && (si != null))
        {
            PageTemplateInfoProvider.RemovePageTemplateFromSite(pti.PageTemplateId,
        }
```

7.2.4.20.2.4 Managing page template scopes

The following sample code shows how you can create a new page template scope and assign it to a template:

[C#]

```
using CMS.PortalEngine;
using CMS.CMSHelper;

...

        // Get template object
        PageTemplateInfo pti = PageTemplateInfoProvider.GetPageTemplateInfo("TestPag

        // If template exists
        if (pti != null)
        {
            // Set page template so that it isn't available for all pages
            pti.PageTemplateForAllPages = false;

            // Create new template scope
            PageTemplateScopeInfo ptsi = new PageTemplateScopeInfo();

            // Set scope properties
```

```
            ptsi.PageTemplateScopeTemplateID = pti.PageTemplateId;
            ptsi.PageTemplateScopeSiteID = CMSContext.CurrentSiteID;
            ptsi.PageTemplateScopePath = "/Home";
            ptsi.PageTemplateScopeLevels = "/{0}/{1}";

            // Save scope to database
            PageTemplateScopeInfoProvider.SetPageTemplateScopeInfo(ptsi);

            // Update page template
            PageTemplateInfoProvider.SetPageTemplateInfo(pti);

        }
```

The following sample code shows how you can delete a page template scope from the system:

[C#]

```
using CMS.PortalEngine;

...

        // Get template scope by ID
        PageTemplateScopeInfo ptsi = PageTemplateScopeInfoProvider.GetPageTemplateSc

        // If exists
        if (ptsi != null)
        {
            // Delete from database
            PageTemplateScopeInfoProvider.DeletePageTemplateScopeInfo(ptsi);
        }
```

7.2.4.20.3 Page layouts

7.2.4.20.3.1 Getting page layout data

The following sample code shows how you can get a pre-defined page layout as a *LayoutInfo* object:

[C#]

```
using CMS.SiteProvider;

...

        // Get layout object by layout code name
        LayoutInfo li = LayoutInfoProvider.GetLayoutInfo("LayoutCodeName");

        // Get layout object by layout ID
        LayoutInfo li2 = LayoutInfoProvider.GetLayoutInfo(1);
```

The following sample code shows how you can get a DataSet containing pre-defined page layouts from the system:

[C#]

```
using System.Data;
using CMS.SiteProvider;

...

        string where = "LayoutID > 10";
        string orderby = "LayoutCodeName";
        int topN = 10;
        string selectedColumns = "LayoutCodeName,LayoutCode";

        // Get dataset of page layouts according to the given parameters
        DataSet ds = LayoutInfoProvider.GetLayouts(where, orderby);

        DataSet ds2 = LayoutInfoProvider.GetLayouts(where, orderby, topN, selectedCo
```

7.2.4.20.3.2 Managing page layouts

The following sample code shows how you can create a new pre-defined page layout:

[C#]

```
using CMS.SiteProvider;

...

        // Create new layout object
        LayoutInfo li = new LayoutInfo();

        // Set properties
        li.LayoutDisplayName = "Test layout";
        li.LayoutCodeName = "TestLayout";
        li.LayoutDescription = "This is a layout created for an API Example";
        li.LayoutCode = "<cc1:CMSWebPartZone ID=\"zoneLeft\" runat=\"server\" />";

        // Save to database
        LayoutInfoProvider.SetLayoutInfo(li);
```

The following sample code shows how you can modify an existing pre-defined page layout:

[C#]

```
using CMS.SiteProvider;

...
```

```
        // Get layout object by layout code name
        LayoutInfo li = LayoutInfoProvider.GetLayoutInfo("TestLayout");

        // If layout exists
        if (li != null)
        {
            // Update properties
            li.LayoutDisplayName += " updated";
            li.LayoutCode += " updated";

            // Save to database
            LayoutInfoProvider.SetLayoutInfo(li);
        }
```

The following sample code shows how you can delete a pre-defined page layout from the system:

[C#]

```
using CMS.SiteProvider;

...

        // Get layout object by layout code name
        LayoutInfo li = LayoutInfoProvider.GetLayoutInfo("TestLayout");

        // If layout exists
        if (li != null)
        {
            // Delete from database
            LayoutInfoProvider.DeleteLayoutInfo(li);
        }
```

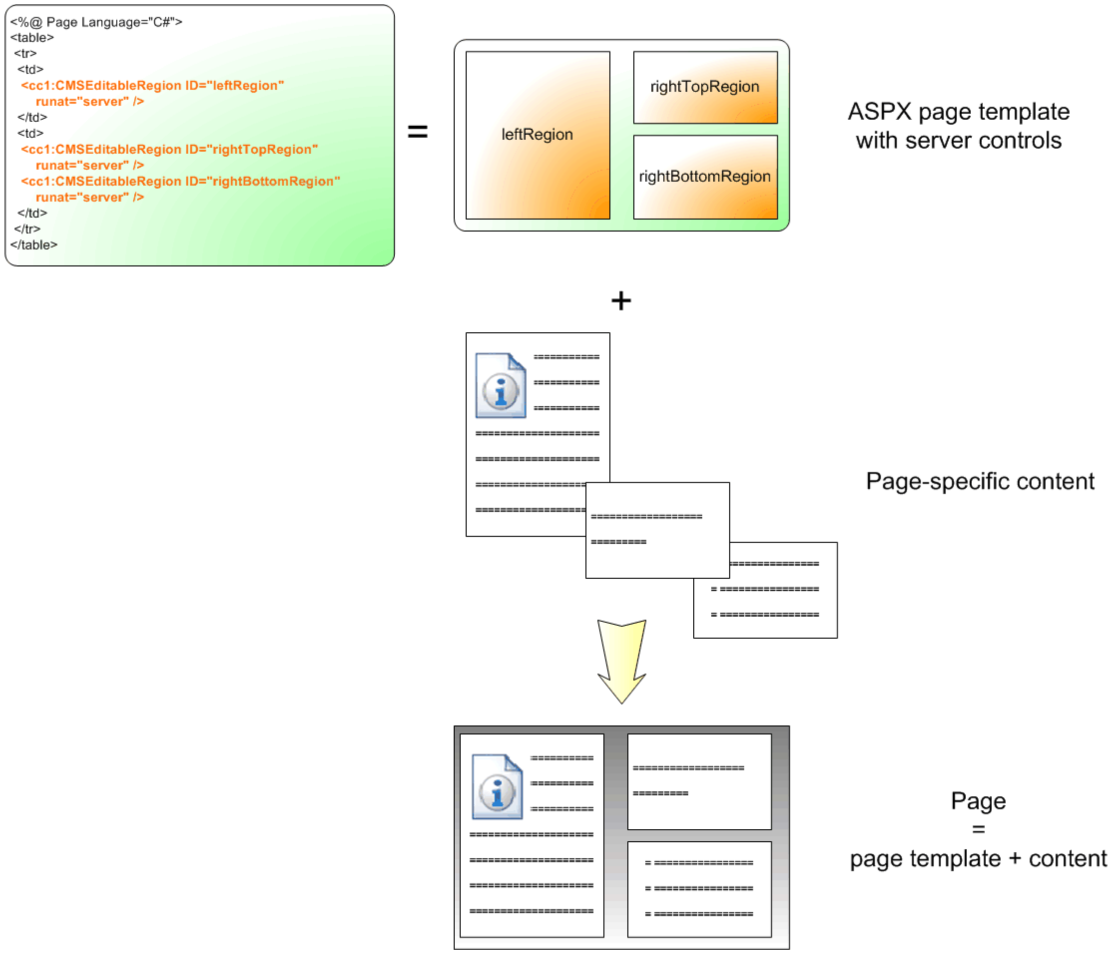
7.2.5 ASPX page template development model

7.2.5.1 Overview

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

What does the ASPX page template consist of?

The page template is a combination of static HTML code and ASP.NET server controls (or user controls) that render dynamic content. You can also use code behind (using either VB.NET or C#) to modify page behavior and add custom functionality. The following figure illustrates how an ASPX page template and page content are combined to display a page:



How is the ASPX page template processed?

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

User requests page
http://localhost/CorporateSite/**Our Company.aspx**

The request is processed by Kentico CMS URL rewriting engine.

The engine recognizes that page /Our Company should be displayed using the SingleArticle.aspx template.

The path is rewritten to:
http://localhost/CorporateSite/SingleArticle.aspx?aliasPath=**/Our Company**

The SingleArticle.aspx page is executed.

The CMSPageManager control gets the aliasPath parameter from the URL and loads content for page **/Our Company**. It then displays the text using appropriate CMSEditableRegion controls. Your custom code is executed as usual.

The output is (optionally) processed by output filter that fixes some form settings and improves XHTML-compatibility.

Final page is displayed to the visitor.

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

As you can see, the system uses standard ASP.NET architecture. If you developed the website without Kentico CMS, you would most likely use URLs like this: **/news.aspx?newsid=127** which is similar to **/news.aspx?aliaspath=/news/your-first-news.aspx** used in Kentico CMS. Kentico CMS also uses friendly URLs in format **/news/your-first-news.aspx** that are better for search engine optimization.

7.2.5.2 Creating a new ASPX page template

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

Adding Kentico CMS Controls to your Visual Studio Toolbox

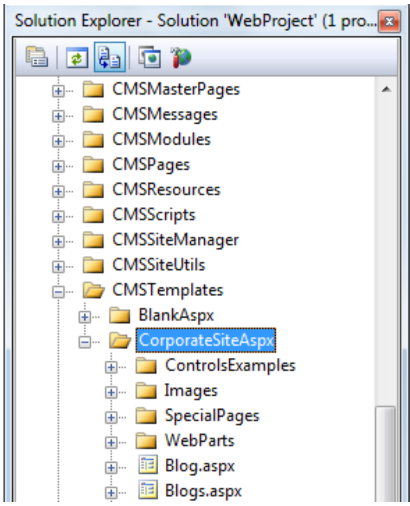
Before you start development of ASPX page templates, it's recommended that you add Kentico CMS Controls to your Visual Studio Toolbox so that you can simply drag and drop the controls onto the ASPX pages.

You can find step-by-step instructions in the Adding Kentico CMS Controls to the

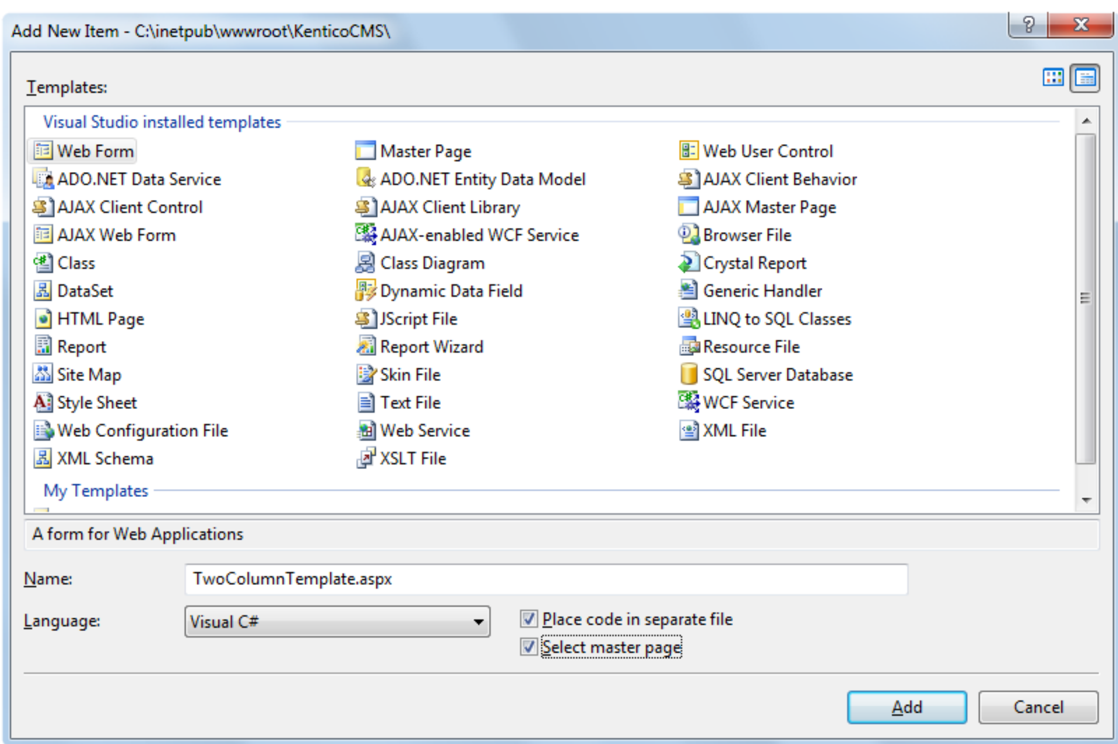
Toolbox topic.

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

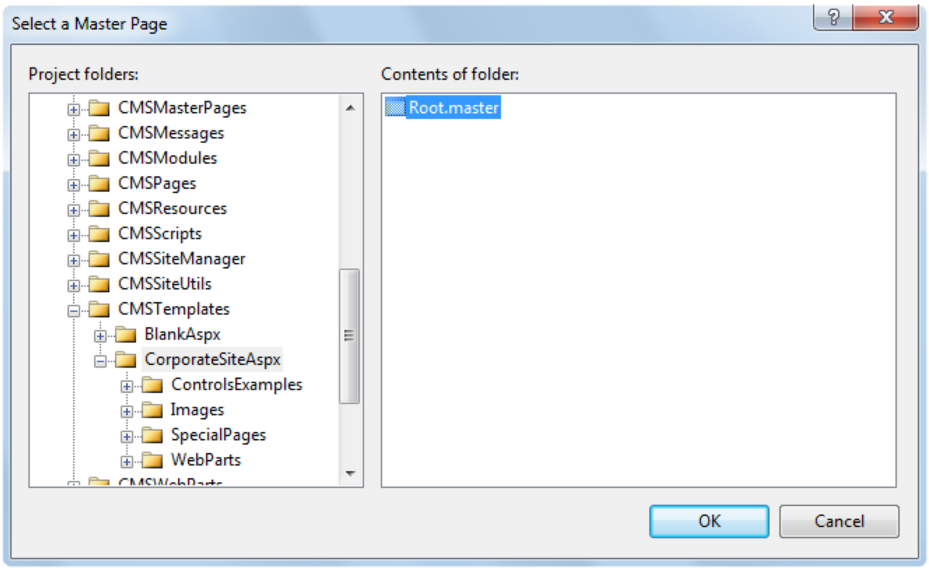
2. Now right-click the **CMSTemplates -> CorporateSiteASPX** folder in the Solution Explorer and select **Add New Item**:



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



4. The **Select a Master Page** dialog appears. Choose the folder **CMSTemplates/CorporateSiteASPX** and choose the **root.master** file and click **OK**.



Writing the ASPX code

5. Switch to the **Source** view of the newly created ASPX page. Add the following line under the **<%@ Page %>** directive:

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

Then add the following code inside the **<asp:Content></asp:Content>** control:

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

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

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

6. Switch to the code behind. You need to add a reference to the **CMS.UIControls** namespace:

[C#]

```
using CMS.UIControls;
```

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

[C#]

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

to this:

[C#]

```
public partial class CMSTemplates_CorporateSiteAspx_TwoColumnTemplate :
TemplatePage
```

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

Please keep in mind that the name of the class must be identical to the value of the **Inherits** attribute of the **<%@ Page %>** directive on the ASPX page. This is case sensitive.

Registering the ASPX page as a page template

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

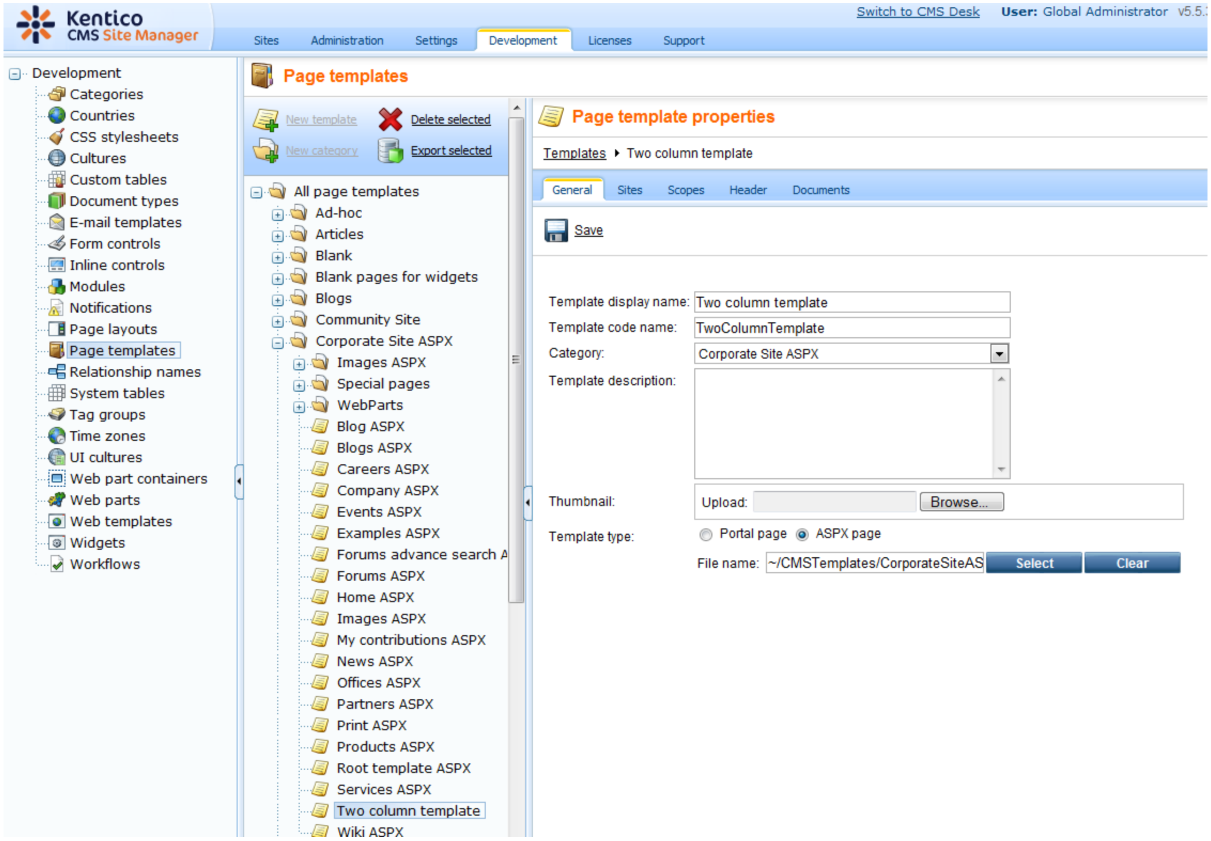
8. Sign in to **Site Manager** and go to **Development -> Page templates**. Click the **Corporate Site ASPX** folder and click **New template**. Enter the following values:

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

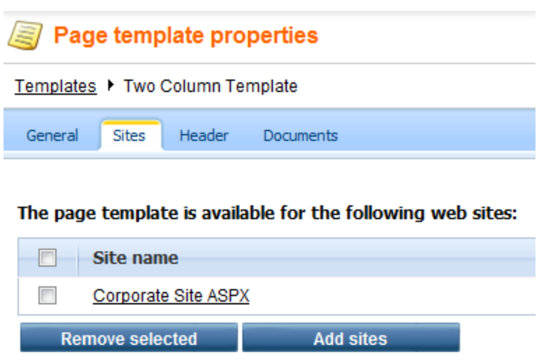
Click **OK**. Now enter the following value in the **File name** field:

~/CMSTemplates/CorporateSiteASPX/twocolumntemplate.aspx

It is the virtual path of our ASPX page. Alternatively, the **Select** button can be used to manually select the file. Click **Save**.

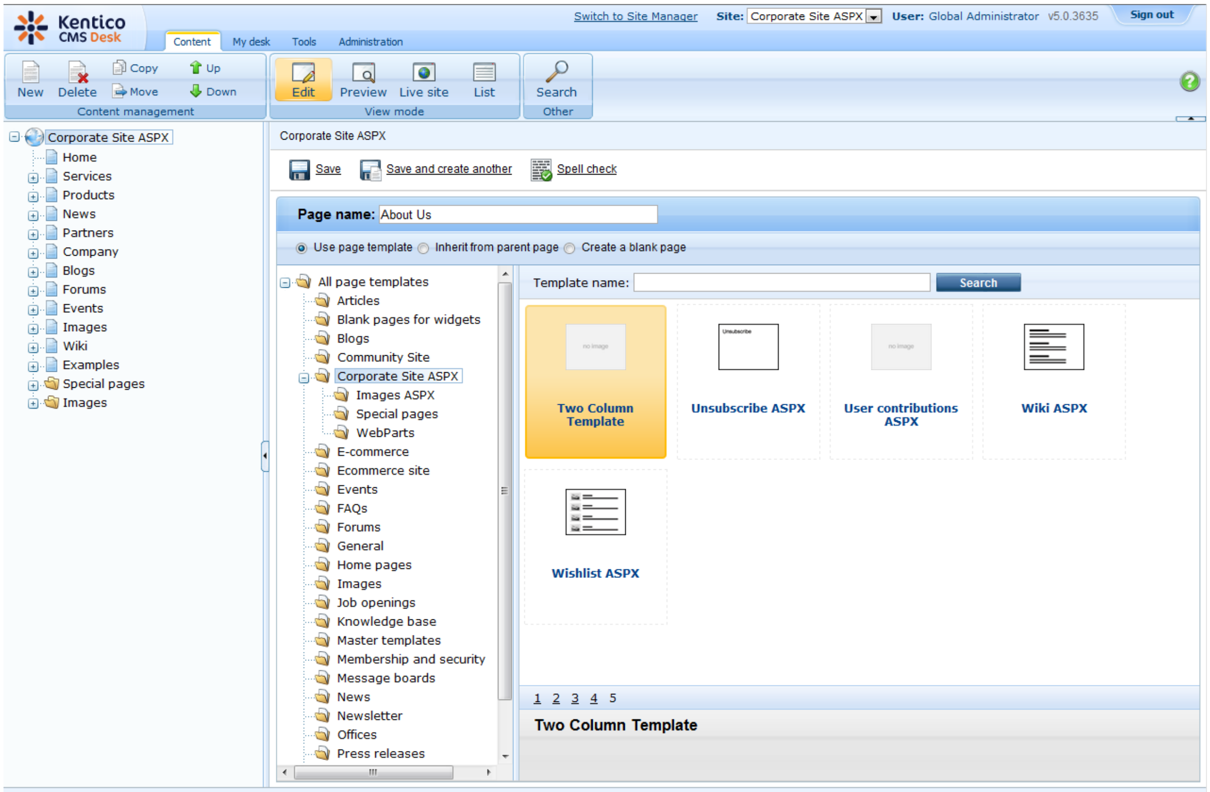


9. Now switch to the **Sites** tab, assign the page template to the current website using the **Add sites** button and click **OK**.

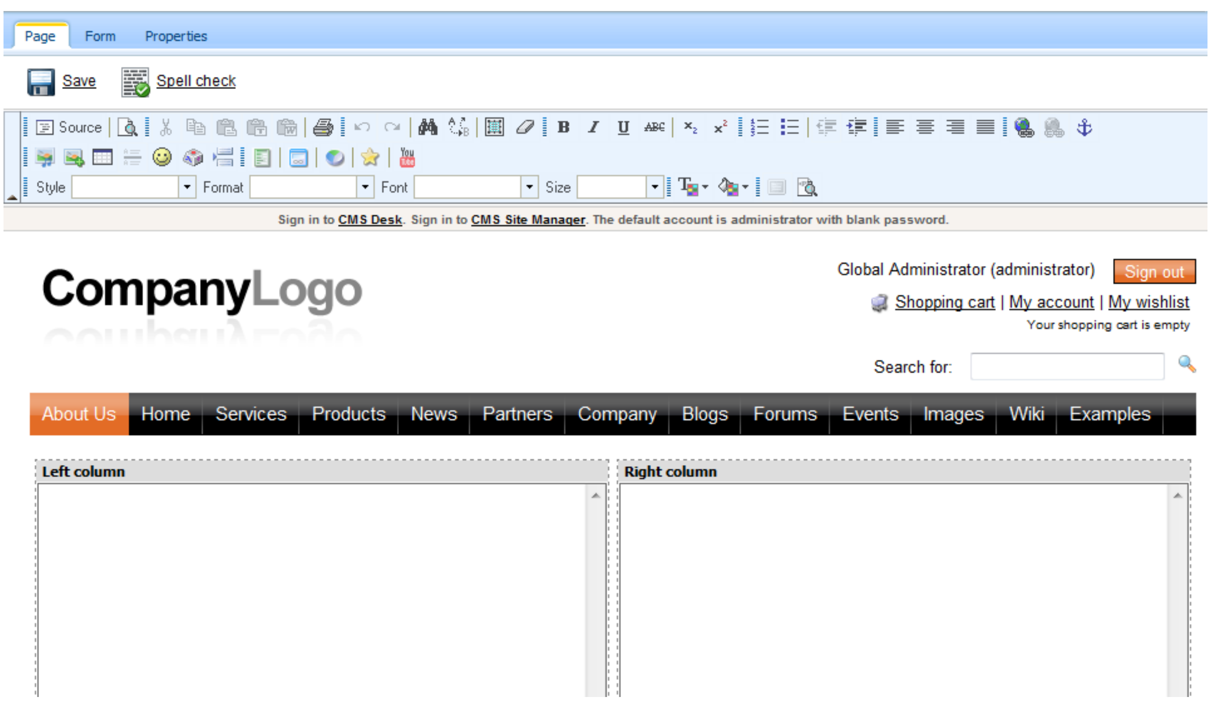


Creating an About Us page based on the new page template

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



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



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

7.2.5.3 Creating ASPX master pages

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

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

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

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

The following code sample defines a very simple master page.

Please note: If you installed the Kentico CMS project as a web application, you need to rename the *CodeFile* property on the first line to *Codebehind* for the code example to be functional.

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Root.master.cs" Inherits
="CMSTemplates_CorporateSiteASPX_Root" %>
<%@ Register Assembly="CMS.PortalControls" Namespace="CMS.PortalControls" TagPrefix=
<%@ Register Assembly="CMS.Controls" Namespace="CMS.Controls" TagPrefix="cms" %>
<%=DocType%>

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <asp:literal runat="server" id="ltlTags" enableviewstate="false" />
</head>
<body class="<%=BodyClass%>" <%=BodyParameters%>>

    <form id="form2" runat="server">

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

    </form>
</body>

</html>
```

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

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

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

In the code behind file, you need to add a reference to the CMS.UIControls namespace:

[C#]

```
using CMS.UIControls;
```

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

[C#]

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

And you also need do add the following code to the master page code behind class:

[C#]

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

    this.PageManager = this.CMSPageManager1;
}

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

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

[VB.NET]

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

    Me.PageManager = Me.CMSPageManager1
End Sub

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

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

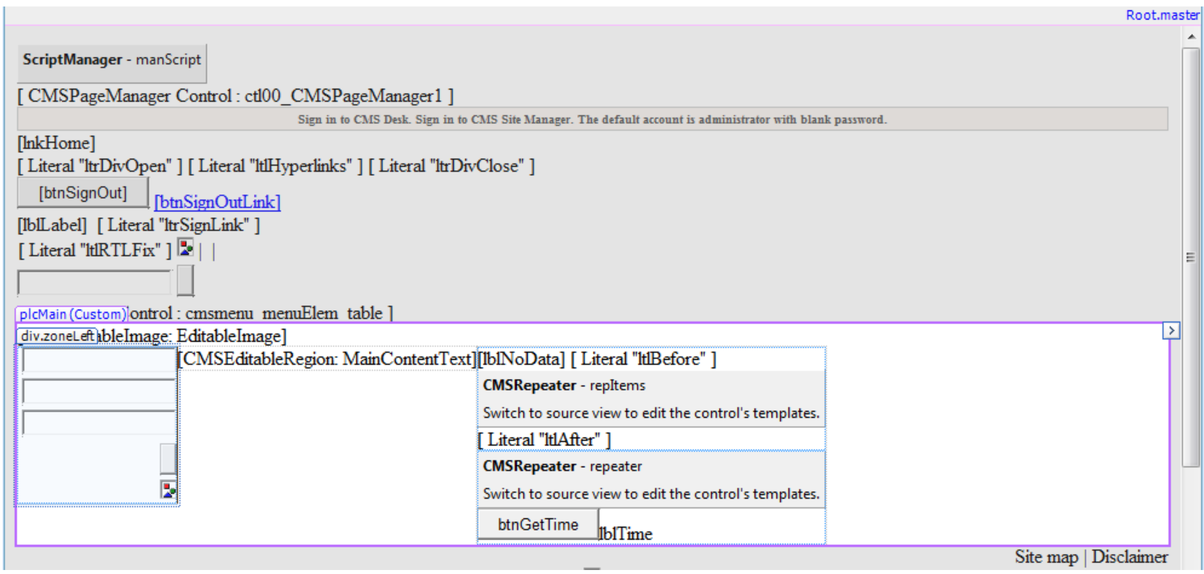
You should store master pages in the **CMSTemplates** folder together with page templates, so that they are exported with your website.

7.2.5.4 Adding custom code to the ASPX page template

In this simple example, you will see how you can easily add custom code to the Home page template. You will see that you can add custom code in Visual Studio, as you usually do. You will need to use the sample **Corporate Site ASPX** website for this example.

1. Open the web project in Visual Studio using the **WebProject.sln** file and open the **home.aspx** page located in the **CMSTemplates\CorporateSiteASPX** folder.

2. Switch to the **Design** tab and add a new button to the page. Call it **btnGetTime** and set its text to **Get time**. Add a new label, call it **lblTime** and clear its text.

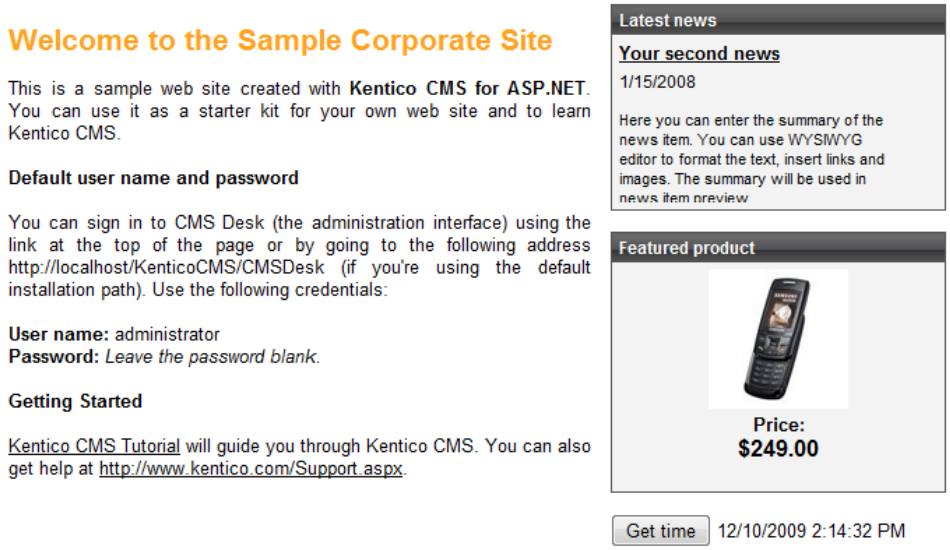


3. Double-click the button and add the following code inside the click event handler:

[C#]

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

4. Save the changes and view the home page on the live site. When you click the button, you can see the label displays the current date and time:



As you can see, you can use the standard programming methods you usually use. You can also use the standard debugging process in Visual Studio.

7.2.5.5 Combining ASPX templates and portal engine templates

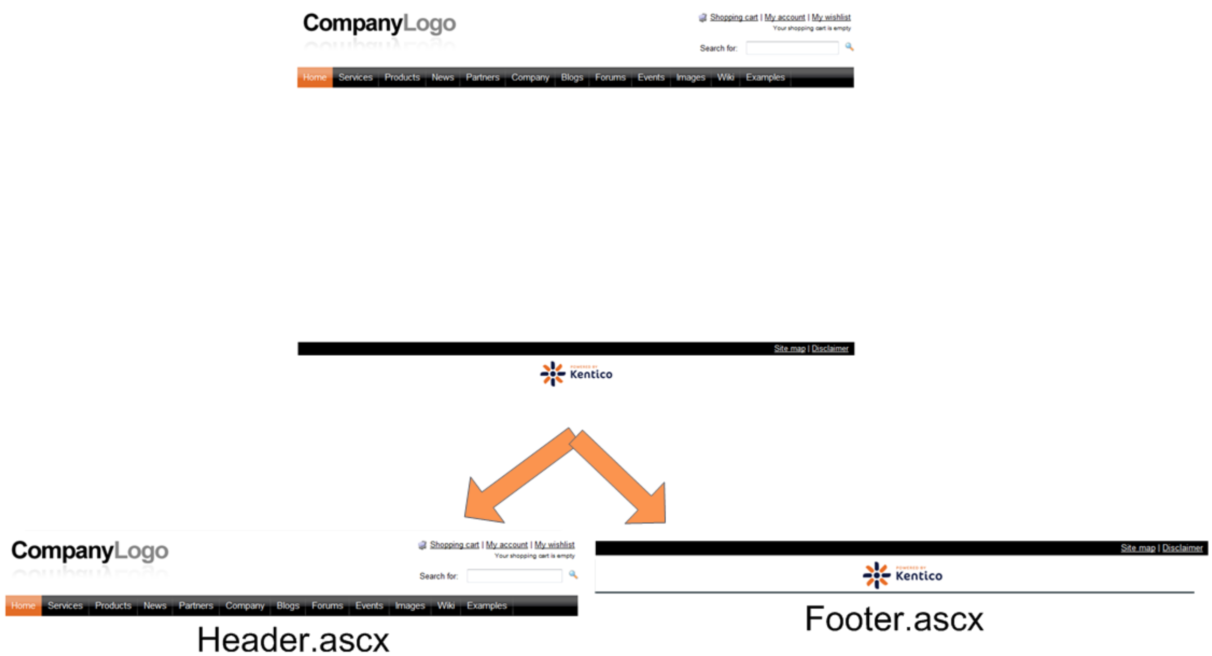
In some cases, you may need to combine ASPX page templates or external ASPX pages and the portal engine page templates. The following text describes how you can accomplish this.

Imagine that you have created a website using the portal engine model - this means that the master page and the other pages use portal engine page templates. The problem here is how to "insert" ASPX pages inside the same master page. Unfortunately, it's not possible to do this directly. However, you can use the following workaround:

Use two master pages - the portal engine master page defined in the root document of the content tree and the ASPX master page (.master file) used for all ASPX pages within the website.

The drawback is that you will need to manage the master page in two places. But this issue also has a simple solution:

Take the web parts and HTML code you have "above" the **Page placeholder** web part in your original master page and place them into a user control called **Header.ascx**. Then, take the web parts and HTML code "below" the **Page placeholder** and place them into a user control called **Footer.ascx** as shown in the following figure:



Now delete the content of the original **portal engine master page** and place only the following web parts onto the master page template:

- **General/User control** web part using the **Header.ascx** user control
- **Page placeholder** web part
- **General/User control** web part using the **Footer.ascx** user control

In the **ASPX master page** (.master file), you simply put:

- the **Header.ascx** user control
- the **ContentPlaceHolder** control
- the **Footer.ascx** user control

In this way, you ensure that you can manage the header and footer in a single place for both the portal engine and ASPX master page.

7.2.5.6 Integration with your existing ASP.NET application

If you need to integrate some existing ASP.NET application with a Kentico CMS website, there are several issues you need to consider. This chapter contains a summary of this topic, if you need more details or help with some particular issues, please contact Kentico support.

Location of CMS and your application

There are three ways how you can organize the CMS web project and your application web project:

1. Mixing both together

It makes sense to mix both applications into a single project if you wish to share functionality, content, security information and session or application variables between the CMS and your application. The easiest way is to use the Kentico CMS web project, as the main project since it's already correctly configured for the CMS, and add your own ASPX pages and other files to this project.

If you need to display your own ASPX pages inside the context of the website, you can simply register them as page templates and then create new pages based on these page templates in the standard website navigation (in the content tree). You will also need to modify your ASPX pages so that they use the master page (.master file) of the Kentico CMS website.

If you wish to use a website built using the Kentico CMS portal engine development model, please also read the Combining ASPX templates and portal engine templates topic.

2. Having the CMS in the root and your application in a sub-folder

If your application can or needs to run separately from the CMS and you want the CMS to manage the main website, you can place the Kentico CMS web project in the root of the website and your application into a sub-folder. You will need to create a virtual directory for the sub-folder so that your application runs correctly.

3. Having your application in the root and the CMS in a sub-folder

If your application can or needs to run separately from the CMS and your application is the main part of the website and you wish to use the CMS only for some sub-section of the website where you publish the content, you need to place the CMS into a sub-folder and create a virtual directory for it.

Interaction between the CMS and your application

If you need to include your application inside the website (front-end), you can do so either through ASPX pages (see paragraph **Mixing both together** above) or you need to create ASCX user controls that you place into the CMS website.

If you need to build an application that will interact with the CMS system, but will mostly provide back-end user interface, you can create a custom module as described in the Custom modules topic.

Sharing security information between the CMS and your application (single-sign-on)

If you wish to use single-sign-on for both your application and the CMS, you need to configure your environment as described in the Single sign-on topic. If you wish to use a single system of permissions (authorization), you can leverage the permission system for custom modules as described in the Custom modules topic.

7.2.5.7 Displaying data from an external database

Besides displaying Kentico CMS content, you can also display data from your external database or Web Service. In this case, you need to add custom code to your ASPX page that will use ADO.NET to retrieve the data or that will contact the Web Service and call its methods. Since you can place any custom code into the page (page template), you will simply use the standard ASP.NET code you would use when creating the website from scratch.

Example: Retrieving data from the sample Northwind database

In this simple example, you will see how to display data on the sample Corporate Site ASPX website from the *Categories* table of the Northwind database using ADO.NET. You may need to use some other database if you do not have the sample database Northwind on your server.

Open the web project in Visual Studio using the **WebProject.sln** file. Open the *CMSTemplates/CorporateSiteASPX/Home.aspx* page.

Drag and drop the standard ASP.NET **GridView** control onto the page and set its ID to **GridView1**.

Add the following line to the beginning of the code behind file:

[C#]

```
using System.Data.SqlClient;
```

Add the following code to the **Page_Load** method:

[C#]

```
// create sql connection - you could use Oracle or OLEDB provider as well
SqlConnection cn = new SqlConnection("Persist Security Info=False;
database=northwind;server=server1;user id=sa;password=psswd;Current
Language=English;Connection Timeout=120;");
// create data adapter
SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM categories", cn);
DataSet ds = new DataSet();

// fill the dataset with data from database
da.Fill(ds);

// bind data to the grid view
GridView1.DataSource = ds;
GridView1.DataBind();
```

Save all changes. Look at the page on the live website. You will see a grid with data from the external database:

CategoryID	CategoryName	Description
1	Animals	Dogs, cats, horses
2	Colors	Red, green, blue, orange,...
3	Names	John, Kevin, Julia, Robin

As you can see we used standard ASP.NET methods to display external data on the website.

7.3 Caching and performance

7.3.1 Overview

The performance of your website depends on many aspects:

1. Hardware on which your website and database server are running.
2. Available system performance when you're sharing system resources with other applications (typically in shared hosting environment).

3. Size of your website.
4. Complexity of the website (number of nesting levels, number of web parts on a page, etc.)
5. Custom code added to the website.
6. Use of caching.
7. Other special circumstances, such as network connectivity between web server and SQL server, etc.

7.3.2 Caching options

Global caching settings

Go to **CMS Site Manager -> Settings -> website**, choose your website from the drop-down list. You can configure the following values:

- **Cache content (minutes) -** this option specifies that all web parts/controls should cache the content they retrieve from Kentico CMS. You can override this value by setting the Cache minutes property of the web parts to 0, which disables caching for the given control, or generally to some different number of minutes. It's recommended that you cache all possible content that is not modified too often. The drawback of this option is that when you modify some content, the changes appear on the live site after the old version expires in the cache.

- **Cache images (minutes) -** this option is used only for caching of images. It's recommended that you always use it. Kentico CMS automatically removes the cached image when it's modified, so it doesn't cause displaying of outdated content.

- **Cache page info (minutes) -** this option is used for caching of page content and metadata. Since Kentico CMS often retrieves page information many times during a single page processing, it's actually a must to **always set this value to at least 10 minutes!** Kentico CMS automatically removes the cached page when it's modified, so it doesn't cause displaying of outdated content.

Full-page caching

Full-page caching represents the most powerful option. It caches the whole page, so it's not necessary to contact the SQL Server and run the page code again when the page is requested second time. You can configure the full-page caching in **CMS Desk -> Properties -> General**. The configuration is automatically **inherited to child pages** unless you disable caching on them.

The page stored in the cache is automatically removed when you modify the given page. However, if the page displays other documents (such as news list) and you modify these documents, the page will not be updated.

This option is not suitable for pages with web parts that need to be refreshed very often (e.g. the Random document web part) since you cannot disable caching for particular web parts. For such pages, it's recommended that you do not use full-page caching and use content caching instead.

Full client cache

You can enable full client cache by adding the following key into the *configuration/appSettings* section of your *web.config*:

<add key="CMSFullClientCache" value="true"/>

If you enable this type of caching, **CSS styles** and **images** will be cached in the client browser. This type of caching can be useful only for live site displaying, as changes made to the page in non-live-site modes would not be displayed due to the caching.

Web part/control-level caching settings

Some web parts/controls used for displaying content have two properties related to caching:

- **Cache item name/CacheItemName -** this property specifies the key name under which the content will be stored in the cache; if not specified, the system generates the key name automatically based on the site name, page path, web part ID and current user name.

- **Cache minutes/CacheMinutes -** this property specifies for how long the web part should cache the content retrieved from Kentico CMS. The default value is defined in the **Cache content (minutes)** site settings value described above. You can override the global value by setting this property to a different number.

- **Cache dependencies/CacheDependencies -** using this property, you can specify which object changes cause the web part's cache to get cleared. Below, you can find a table showing which dummy cache keys get touched when some object gets changed, including some examples. By entering the appropriate dummy keys, one per line, you can specify that when the object gets changed, the cache gets cleared.

  If you check the **Use default cache dependencies** check-box, default settings will be used. The default settings are configured for each web part and include all possible object changes that the content of the web part could depend on.

Object	**Touched keys**	**Sample values**
Document (TreeNode)	node|<sitename>|<aliaspath>|<culture> node|<sitename>|<aliaspath> nodeid|<nodeid> nodeid|<linkednodeid> nodes|<sitename>|<classname>|all **+ for every parent node**: node|<sitename>|<aliaspath>|childnodes	node|corporatesite|/home|en-us node|corporatesite|/home nodeid|12 nodeid|34 nodes|corporatesite|cms.menuitem|all node|sitename|/|childnodes
Any object (except documents)	<classname>|all <classname>|byid|<id> <classname>|byname|<codename> <classname>|byguid|<guid>	cms.user|all cms.user|byid|53 cms.user|byname|administrator cms.user|byguid|1ced44f3-f2fc- …
Metafile	metafile|<guid>	metafile|1ced44f3-f2fc- …
Document attachment	attachment|<guid>	attachment|1ced44f3-f2fc- …
Forum attachment	forumattachment|<guid>	forumattachment|1ced44f3-f2fc- …
Avatar	avatarfile|<guid>	avatarfile|1ced44f3-f2fc- …
Media file	mediafile|<guid> mediafile|preview|<guid>	mediafile|1ced44f3-f2fc- … mediafile|preview|1ced44f3-f2fc- …

Page template	template|<id>	template|12
CacheHelper .ClearFullPageCache	fullpage	fullpage

Example 1: let's presume that you have a web part displaying some information about users. Therefore, whenever some user gets its details modified, the web part's cache should be cleared. To ensure this, you need to enter **cms.user|all** into the field, which is the dummy key that would get touched whenever some user's details get changed.

Example 2: now let's presume that your web part is displaying information about one particular user - the administrator. Her user name is *administrator*, her ID is *53* and her GUID is something beginning with *1ced44f3-f2fc*. So if you want to have the cache cleared whenever this user's details are changed, you can use any of the following three keys that specify the user by the previously named properties:

- **cms.user|byid|53**
- **cms.user|byname|administrator**
- **cms.user|byguid|1ced44f3-f2fc-...**

Partial caching

Partial caching is, simply put, full-page caching for web parts. Like with full-page caching, partial caching also stores the output HTML code. But in this case, it doesn't store the whole page, but only the output HTML code of the particular web part. The web parts have the **Partial cache minutes** and **Partial cache dependencies** properties, which can be used the same way as the **Cache minutes** and **Cache dependencies** properties described above.

Please note: default partial cache dependencies are not specified for the web parts, so if you want to use this feature, you need to specify them manually by entering the dummy keys as described above.

Caching and personalization

If your website contains sections for site members, the caching will be personalized, which means each signed in user will have their own cache. This may lead to large memory consumption, so it's recommended that you set caching to lower values for membership sites.

Previewing modifications made to pages with caching

Caching may confuse the content editors since they will not see the changes they made on the live site immediately. In such case, they can preview their changes in the **Preview** mode in Kentico CMS Desk since this mode doesn't use caching. The **Edit** mode doesn't use caching either.

7.3.3 Troubleshooting performance issues

If you encounter performance issues, please try to follow these steps to make sure that your system is optimized for best performance:

1. **Make sure you're using the latest version of Kentico CMS**
   We improve the performance with every new release. Especially the 2.0 and 2.1 versions didn't provide a very good performance. You can find the version number in the lower right corner of Kentico CMS Desk -> Content dialog or on the logon screen of the administration interface.

2. **Make sure caching is configured on your website.**
   Go to CMS Site Manager -> Settings, choose the appropriate website in the Sites drop-downlist and choose the website category. Make sure the values are set like these:
   - Cache content >= 0
   - Cache images > 0, at least 10 minutes recommended (images are automatically removed from the cache and reloaded if you modify them)
   - **always set Cache page info > 0**, at least 10 minutes recommended (page data is automatically removed from the cache and reloaded if you modify it)

3. **Try configuring full-page caching**
   Full-page caching is the most powerful caching option. Once the page is cached in the memory during the first view, it's displayed without contacting SQL server and running the page code. You can configure full-page caching in **CMS Desk -> Content -> Properties -> General -> Cache**.

   The page is automatically removed from the memory and reloaded when you modify it's content. Please note that the performance improvement will be visible only during the second load of the page.

4. **Turning off output filters**
   In special cases, the website may be slowed down by output filters. Go to Site Manager -> Settings, choose the appropriate website in the Sites drop-down list and choose the **Output filter** category. You can try to temporarily turn off all output filters by setting the following values to / (slash), which will disable the given filter for the whole website:

   - Excluded output form filter URLs
   - Excluded resolve filter URLs
   - Excluded XHTML filter URL

   If it helps, please contact us and we will help you find a workaround (if possible).

5. **Configure file caching**
   Displaying files stored in Kentico CMS repository may be require lots of CPU time and may harm the overall website performance. Please try to configure these values in the **Site Manager -> Settings -> Files** section (your web application needs to have Modify permissions on the disk):

   - Generate thumbnails: yes
   - Redirect files to disk: yes
   - Store files in file system: yes

6. **Check your code**
   If you integrated any custom .NET code into the website, please make sure it works properly. Please be sure to avoid too many database operations and be sure to close the database connections properly. Try to comment out your code and see if it improves the performance.

7. **Check your hardware**
   It's recommended that your system has at least 1 GB RAM and Pentium 4 or Pentium Core 2 Duo (or similar) processor.

8. **Check the other applications/websites running on the same server**
   Whether you use your own server or shared hosting, make sure that the other applications do not take all the server performance. It's highly recommended that you run Kentico CMS in a **separate application pool** on Windows Server 2003.

9. Setting application pool Idle time-out in IIS
If your website is accessed not very frequently (less that every 20 minutes by default), users may experience long delays on first access to the site. To prevent this, you need to set the **Idle time-out (minutes)** property of the application pool to a higher value. This property can be accessed through:

IIS 6: open **IIS Manager** -> select **<machine>/Application Pools** -> right click the application pool -> select Properties -> switch to the **Performance** tab -> set the **Shutdown worker process after being idle for (time in minutes)** to a higher value or disable the option completely by unchecking the box

IIS 7: open **IIS Manager** -> select **<machine>/Application Pools** -> right click the application pool -> select **Advanced Settings** -> the property is located in the **Process Model** section

If the above mentioned settings don't help, please send us an exported copy of your website and we will try to analyze it.

7.3.4 File management and performance

Files can be stored in the file system (faster) or in database. If you're experiencing problems with slow image viewing, please try to configure the following values in the **Site Manager -> Settings -> Files** section:

- **Generate thumbnails**: yes
- **Redirect files to disk**: yes
- **Store files in file system**: yes

You can find more details on file management in chapter Where the files are stored.

7.4 CSS stylesheets and design

7.4.1 Overview

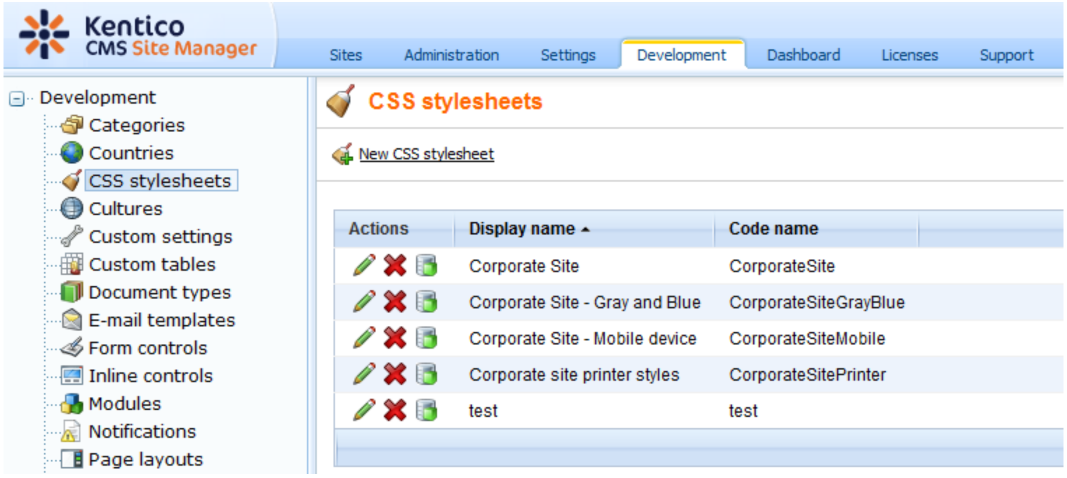
CSS stylesheets allow you to modify the design of your website. You can use standard CSS styles with Kentico CMS as you are used to.

Every website has some **default CSS stylesheet**. It can be modified in **Site Manager -> Sites -> edit ( ) a site**, in the **Site CSS stylesheet** field.

Every page can either use the website stylesheet or it can override it with its own stylesheet. The page stylesheet can be configured in **CMS Desk -> Content -> Properties tab -> General**, in field **CSS stylesheet**.

Creating and managing stylesheets

You can create and manage CSS stylesheets in **Site Manager -> Development -> CSS stylesheets**.



Each stylesheet has the following properties:

- **Stylesheet display name** - name of the stylesheet displayed in the user interface
- **Stylesheet code name** - name of the stylesheet used in website code
- **Stylesheet text** - standard CSS stylesheet code

You also need to enable the stylesheet for particular websites on the **Sites** tab.

Stylesheet URL

You can receive any stylesheet using a URL in the following format:

<domain>/CMSPages/GetCSS.aspx?stylesheetname=mystylesheetcodename

Combining website and page stylesheet

If you need to only combine the website and page stylesheet, you need to configure the page for using its own style sheet and import the website stylesheet using the following CSS directive:

[CSS]

```
@import url(/alfa17/CMSPages/GetCSS.aspx?stylesheetname=corporatesite);
```

The @import directive must be placed at the beginning of the stylesheet. All styles defined after this directive override the styles defined in the imported stylesheet.

Using your favorite CSS editor for stylesheet editing

If you want to edit CSS styles with your favorite editor, you can simply use the @import directive to import your static CSS stylesheet and edit it in your editor. After you finish the design, you can simply copy and paste the stylesheet to Kentico CMS stylesheet.

Using CSS blocks for easier navigation in CSS code

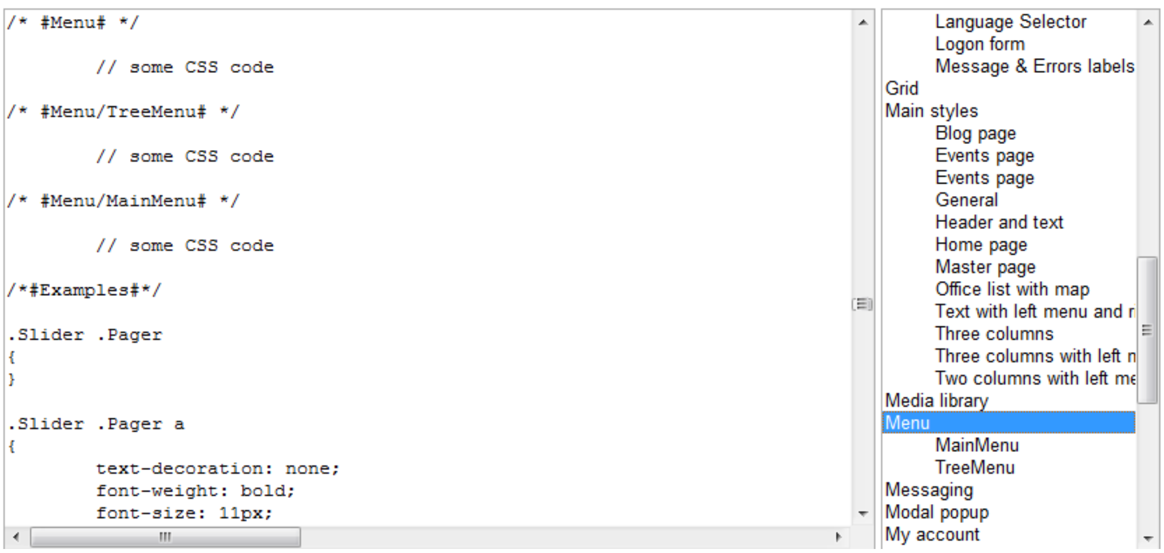
You can use comments in format **`/* #BLOCKNAME# */`** to make your navigation in CSS code easier. The comments may contain sub-blocks separated with a slash, such as **`/* #BLOCKNAME/SUBBLOCK# */`**.

Example:

[CSS]

```
/* #Menu# */

// some CSS code

/* #Menu/TreeMenu# */

// some CSS code

/* #Menu/MainMenu# */

// some CSS code
```

The outlined structure will look like this:



7.4.2 App themes

In some cases, you may leverage the built-in support for ASP.NET themes. You can use them for setting styles of controls that do not have their own CSS class name, such as Datagrid or Calendar.

The name of the theme folder under App_Themes must be same as the code name of the site CSS stylesheet. So if you use the Green stylesheet on your site, your theme must be stored in the App_Themes\green sub-folder.

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

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

Website design files

It's recommended that you store all images or Flash movies that are part of the website design template in the application theme folder. This ensures that the files are exported together with website when your deploying it to some other server.

7.4.3 Printer friendly CSS styles

This chapter explains how to use the printer friendly CSS styles on your website. These styles are used only if a document is sent to a printer.

1. Create a new CSS style sheet in **Site Manager** -> **Development** -> **CSS style sheets**, name it *Printer styles* and set its code name to *Printer_styles*, for example. See the simple example bellow for an illustration of the printer friendly CSS styles created for our default **Corporate site**.

```
.zoneLeft, .zoneRight, .zoneTopInfo, .zoneTop, .horizontalmenu, .zoneBottom
{
       display: none;
}
.eventCalendarDetail .zoneLeft, .eventCalendarDetail zoneRight
{
       display: block;
}
.eventCalendarDetail zoneRight
{
       float: left;
}
.logonReg .zoneLeft, .logonReg .zoneRight
{
       display: block;
}
.logonReg .zoneRight
{
       float: left;
}
.zoneContent
```

```
{
        float: left !important;
}
```

Please note that you have to hide the all the elements that should not be visible in the print version. You can do this by adding **display:none;** to the given style.

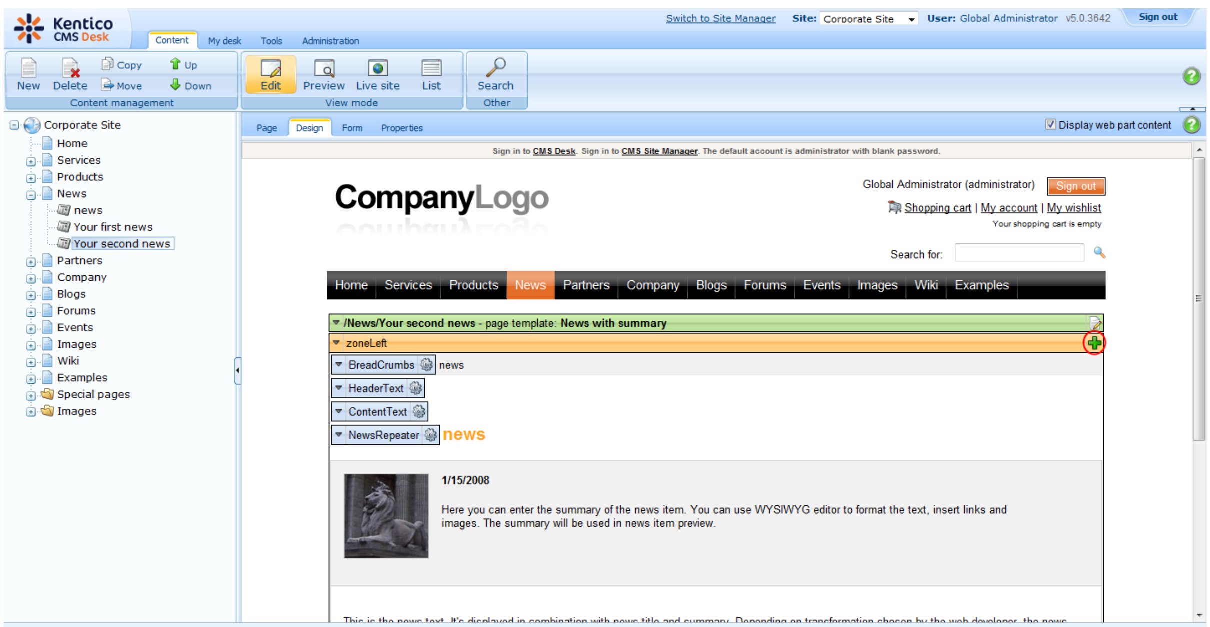
2. Add to the master page header tag link to appropriate CSS style. For example:

```
<link href="CMSPages/GetCSS.aspx?stylesheetname=Printer_styles" type="text/css"
rel="stylesheet" media="print" />
```

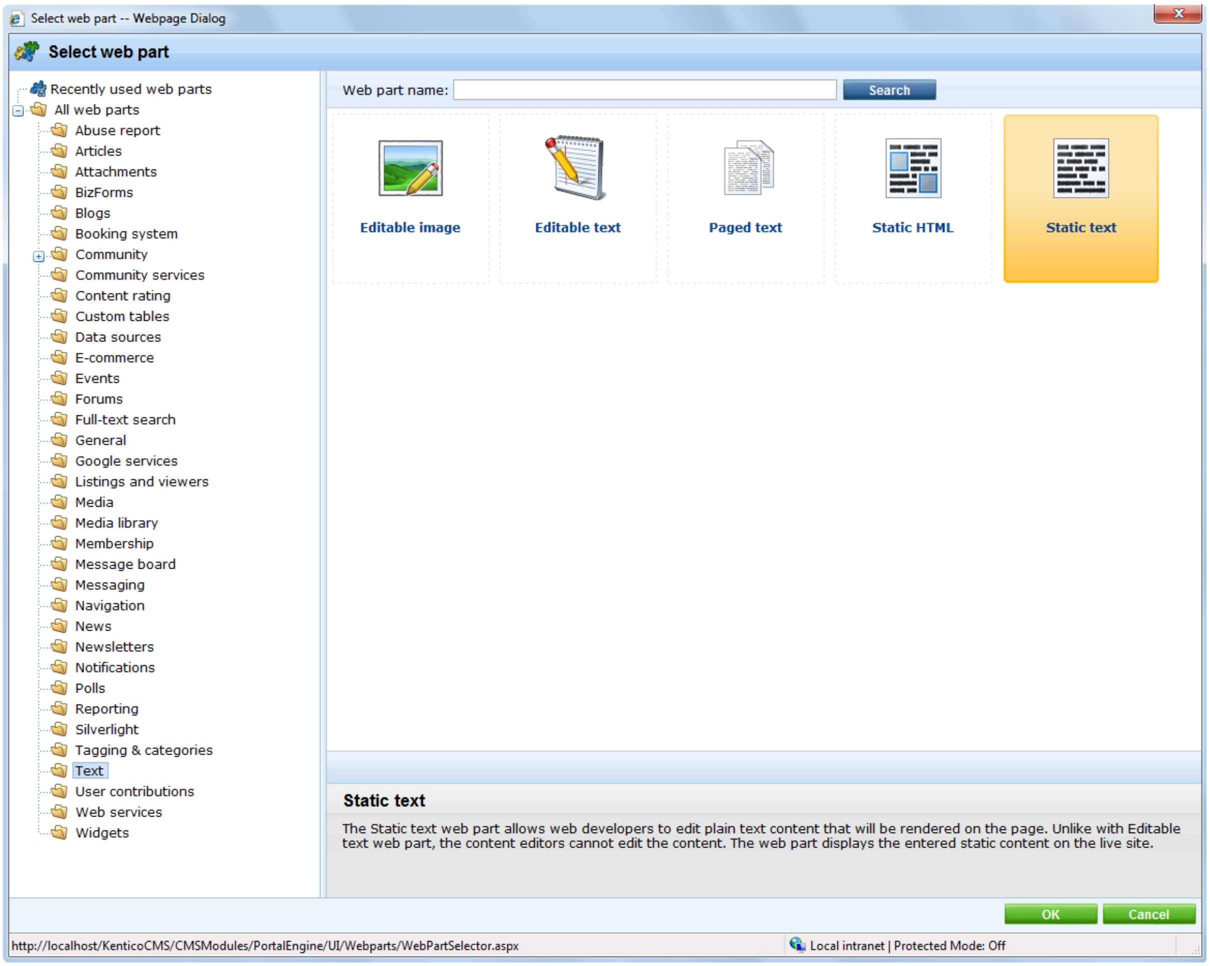
7.4.4 Print page

Kentico CMS allows you to add a link button to your web page that will create print version of the given document. The following example shows you on the sample Corporate Site how to create the given button for the news section.

1. Go to **CMS Desk** -> **Content** -> **News** -> **Your second news** -> **Design** and click the **Add web part** () button at the **zoneLeft**.

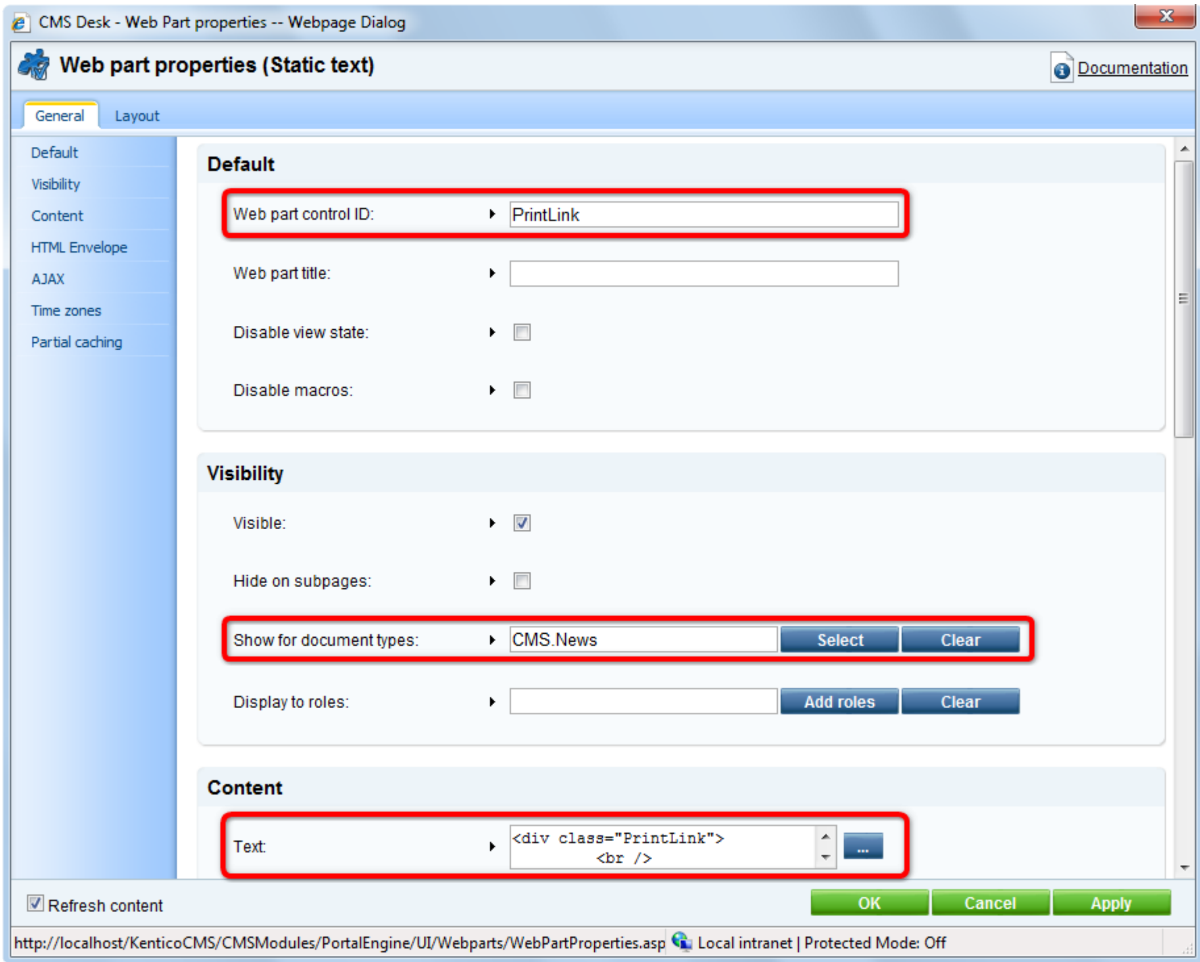


2. Select **Text/Static Text**.

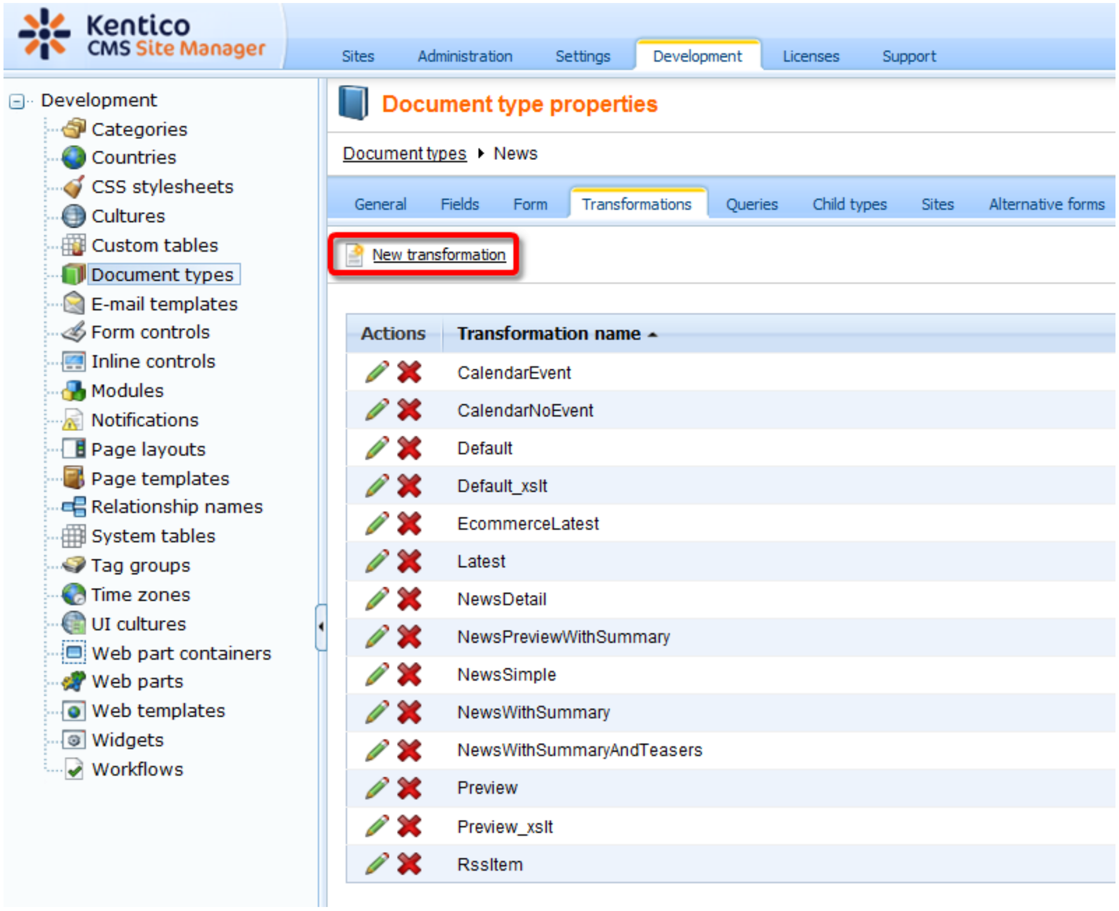


3. In the web part properties, enter *PrintLink* as **ID** and choose **CMS.News** for the Show for document types text box. Enter the following code into the **Text** text box. Then click **OK**.

```
<div class="PrintLink">
       <br />
       <a href="~/SpecialPages/Print.aspx?printpath={%NodeAliasPath%}&classname={%
ClassName%}" target="_blank" >
              <img class="PrintImage" src="~/App_Themes/CorporateSite/Images/
Print.gif" alt="Print" />
              Print
       </a>
</div>
```

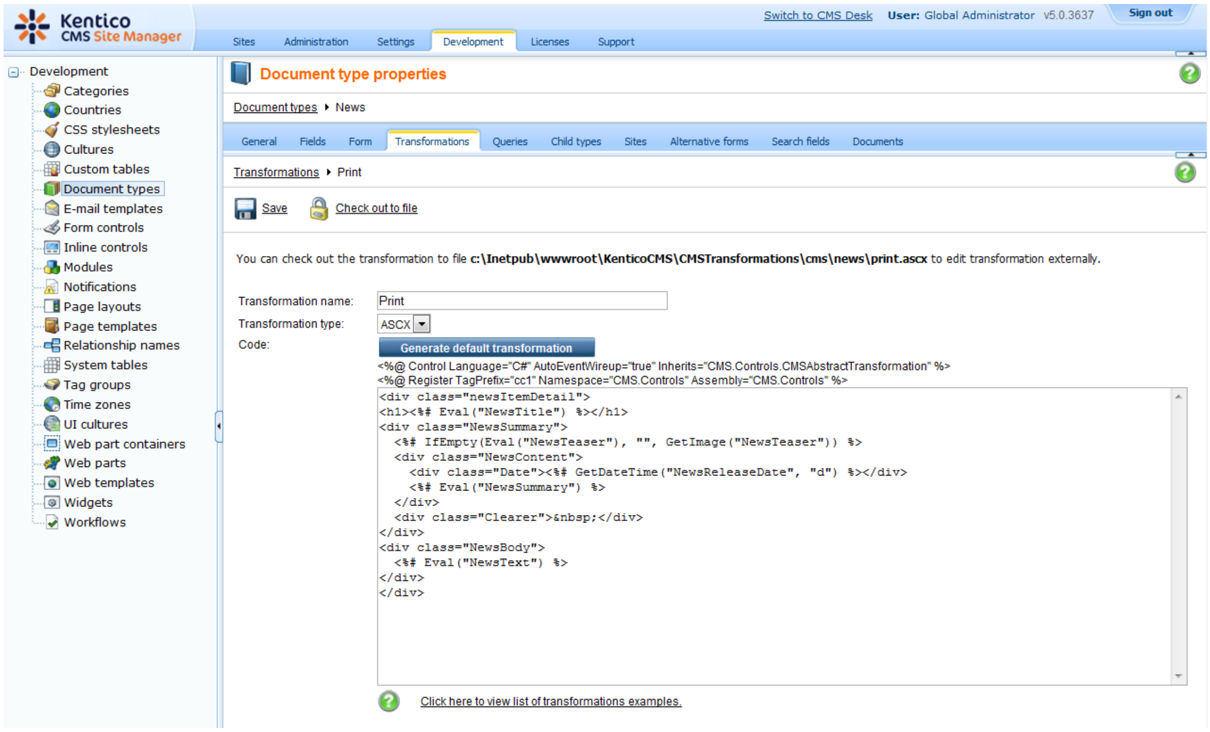


4. Now you have to specify the Print transformation for the new document type. Go to **CMS Site Manager** -> **Development** -> **Document Types** and click the **Edit** button next to the **News** document type. Switch to the **Transformation** tab and click **New Transformation**.

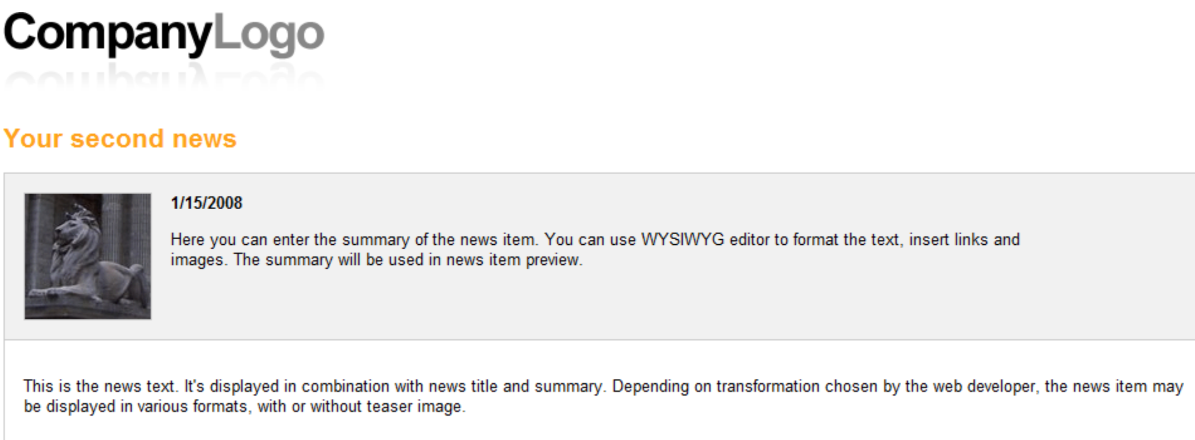


5. Enter *Print* as **Transformation name** and enter the following code into the **Code** text box. Then click **Save**.

```
<div class="newsItemDetail">
<h1><%# Eval("NewsTitle") %></h1>
<div class="NewsSummary">
  <%# IfEmpty(Eval("NewsTeaser"), "", GetImage("NewsTeaser")) %>
  <div class="NewsContent">
    <div class="Date"><%# GetDateTime("NewsReleaseDate", "d") %></div>
    <%# Eval("NewsSummary") %>
  </div>
  <div class="Clearer"> </div>
</div>
<div class="NewsBody">
  <%# Eval("NewsText") %>
</div>
</div>
```



6. Now, go back to **CMS Desk** -> **Content** -> **News** -> **Your second news** and click the newly created **Print** button. You will be redirected to the **Print** page that displays the print version of the given news item.

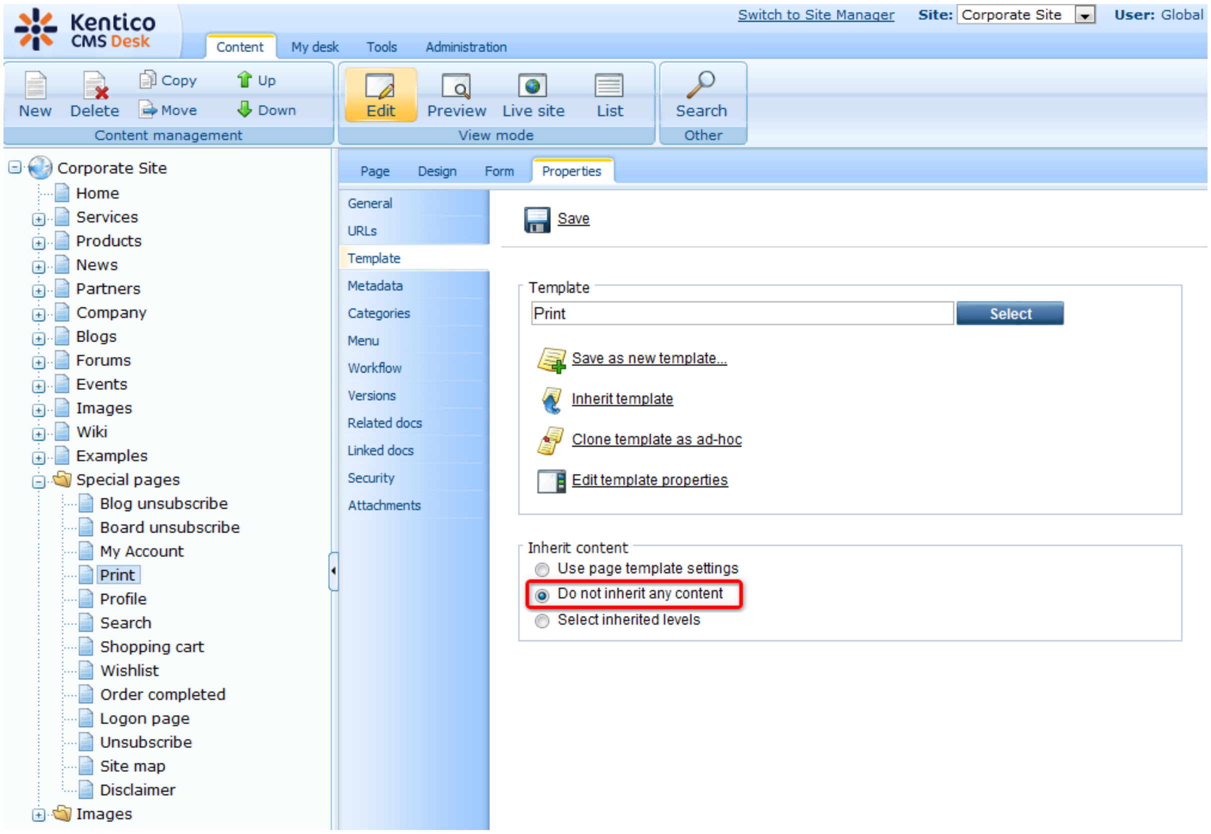


Creating the Print page on your site

The **Print** page is already created on the sample Corporate Site. On your own website, you will have to create it by yourself. The following steps need to be taken in order for the Print page to be created:

1. Add a new blank page to the **Special pages** folder and name it **Print**.

2. Firstly, you have to disable content inheritance at **Properties** -> **Template**.



3. Every **Print** page contains the repeater web part that renders the **Print** transformation for the given document type. The transformation can be specified at **Site Manager** -> **Development** -> **Document Types** -> **<edit () document type>** -> **Transformations**.

Therefore, add the **Repeater** web part and set its properties to the following values:

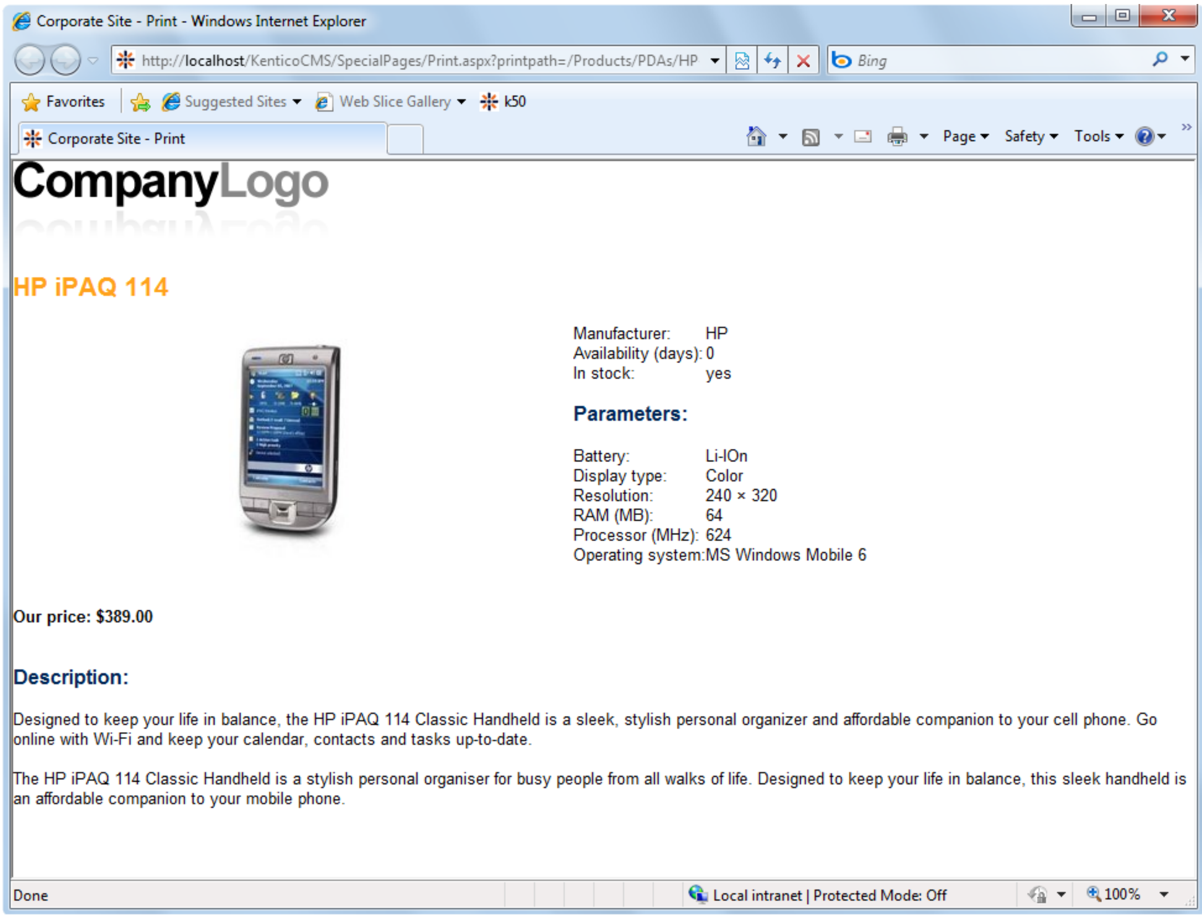
- **Path:** {?printpath|/%?}
- **Document types:** {?classname|cms.root?}
- **Transformation:** {?classname|cms.root?}.print

The **Path** text box specifies the path to the document whose print version you want to make. The value of the **printpath** macro expression is supplied in URL. If no value is supplied in URL (for instance you go directly on the **Print** page from the content tree and not through the print link button), the default **Print** transformation for **cms.root** is displayed.

The **Document types** text box specifies the document types that should be displayed.

The **Transformation** text box specifies the name of the transformation that should be used to render the print version. If the **Print** transformation is not defined for the given document type, the **Print** transformation for the **cms.root** is used.

4. You can try out the functionality of the **Print** page by printing the detail of any product, because the **Print** link button is already created for all products on the Corporate Site.



7.4.5 CSS stylesheet internals and API

7.4.5.1 Database tables and API classes

The following database tables are used to store CSS stylesheets:

- **CMS_CssStylesheet** - contains CSS stylesheets and information about them
- **CMS_CssStylesheetSite** - assigns CSS stylesheets to the sites where they can be used

The CSS stylesheet API is provided by the following **CMS.SiteProvider** namespace classes:

- **CssStylesheetInfo**, **CssStylesheetInfoProvider** - these classes provide functionality for managing CSS stylesheets

The following topics show examples of how these classes can be used:

- Getting CSS stylesheet data
- Managing CSS stylesheets
- Assigning CSS stylesheets to sites

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

7.4.5.2 Getting CSS stylesheet data

The following sample code shows how you can get a CSS stylesheet as a *CssStylesheetInfo* object:

[C#]

```
using CMS.SiteProvider;

...

        // Get stylesheet object by ID
        CssStylesheetInfo csi = CssStylesheetInfoProvider.GetCssStylesheetInfo(1);

        // Get stylesheet object by stylesheet name
        CssStylesheetInfo csi2 = CssStylesheetInfoProvider.GetCssStylesheetInfo(
"CorporateSite");
```

The following sample code shows how you can get a DataSet containing CSS stylesheets:

[C#]

```
using System.Data;
using CMS.SiteProvider;
using CMS.CMSHelper;

...

        string siteName = CMSContext.CurrentSiteName;
        string where = "";
        string orderBy = "";
        string columns = "";
        int topN = 10 ;

        // Get dataset of stylesheets according to the given parameters
        DataSet ds2 = CssStylesheetInfoProvider.GetCssStylesheets(siteName);

        DataSet ds3 = CssStylesheetInfoProvider.GetCssStylesheets(where, orderBy);

        DataSet ds4 = CssStylesheetInfoProvider.GetCssStylesheets(where, orderBy,
topN);

        DataSet ds5 = CssStylesheetInfoProvider.GetCssStylesheets(where, orderBy,
topN, columns);

        // Get dataset of all stylesheet objects
        DataSet ds = CssStylesheetInfoProvider.GetAllCssStylesheets();
```

7.4.5.3 Managing CSS stylesheets

The following sample code shows how you can modify the *Display name* of an existing CSS stylesheet:

[C#]

```
using CMS.SiteProvider;


...

        // Get stylesheet object
        CssStylesheetInfo csi = CssStylesheetInfoProvider.GetCssStylesheetInfo(
"TestStylesheet");

        // If stylesheet exists
        if (csi != null)
        {
            // Update property
            csi.StylesheetDisplayName += " updated";

            // Save to database
            CssStylesheetInfoProvider.SetCssStylesheetInfo(csi);
        }
```

The following sample code shows how you can create a new CSS stylesheet and add it to the system:

[C#]

```
using CMS.SiteProvider;

...

        // Create new object
        CssStylesheetInfo csi = new CssStylesheetInfo();

        // Set properties
        csi.StylesheetDisplayName = "Test stylesheet";
        csi.StylesheetName = "TestStylesheet";
        csi.StylesheetText = "Some CSS code";

        // Save to database
        CssStylesheetInfoProvider.SetCssStylesheetInfo(csi);
```

The following sample code shows how you can delete an existing CSS stylesheet:

[C#]

```
using CMS.SiteProvider;

```

```
...

        // Get stylesheet object
        CssStylesheetInfo csi = CssStylesheetInfoProvider.GetCssStylesheetInfo(
"TestStylesheet");

        // If stylesheet exists
        if (csi != null)
            {
                // Delete from database
                CssStylesheetInfoProvider.DeleteCssStylesheetInfo(csi);
            }
```

7.4.5.4 Assigning CSS stylesheets to sites

The following sample code shows how you can assign a an existing CSS stylesheet to a site:

[C#]

```
using CMS.SiteProvider;

...

        // Get stylesheet and site object
        CssStylesheetInfo csi = CssStylesheetInfoProvider.GetCssStylesheetInfo(
"TestStylesheet");
        SiteInfo si = SiteInfoProvider.GetSiteInfo("CorporateSiteASPX");

        // If both exist
        if ((csi != null) && (si != null))
        {
            // Add stylesheet to site
            CssStylesheetSiteInfoProvider.AddCssStylesheetToSite(csi.StylesheetID,
si.SiteID);

        }
```

7.5 Debugging

7.5.1 Overview

Debugging in the UI

The **Administration -> System -> Debug** tab can be used for system debugging, i.e. finding and fixing performance or settings issues on your website. Debugs are also particularly useful in case that you are reporting an issue to our support department. Every extra bit of information related to your issue that you find in the debugs can significantly shorten the time needed to solve the issue.

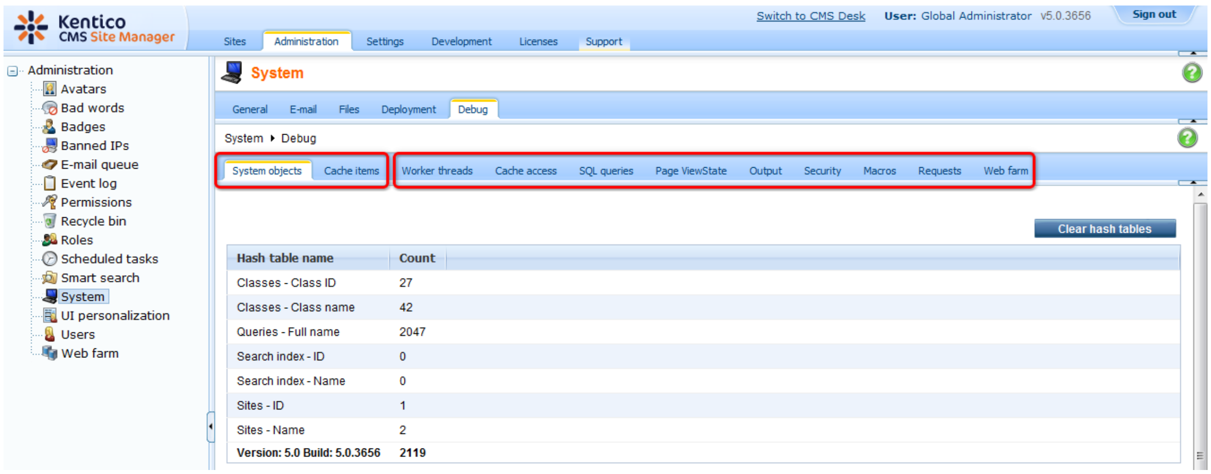
The debugging interface is divided into several sub-tabs. Only the following two tabs are displayed by default. Click the name of the tab to learn more.

- System objects
- Cache items

The other sub-tabs are displayed only after adding a dedicated key into the *AppSettings* section of your *web.config* file. Here again, click the name of the tab to get detailed information.

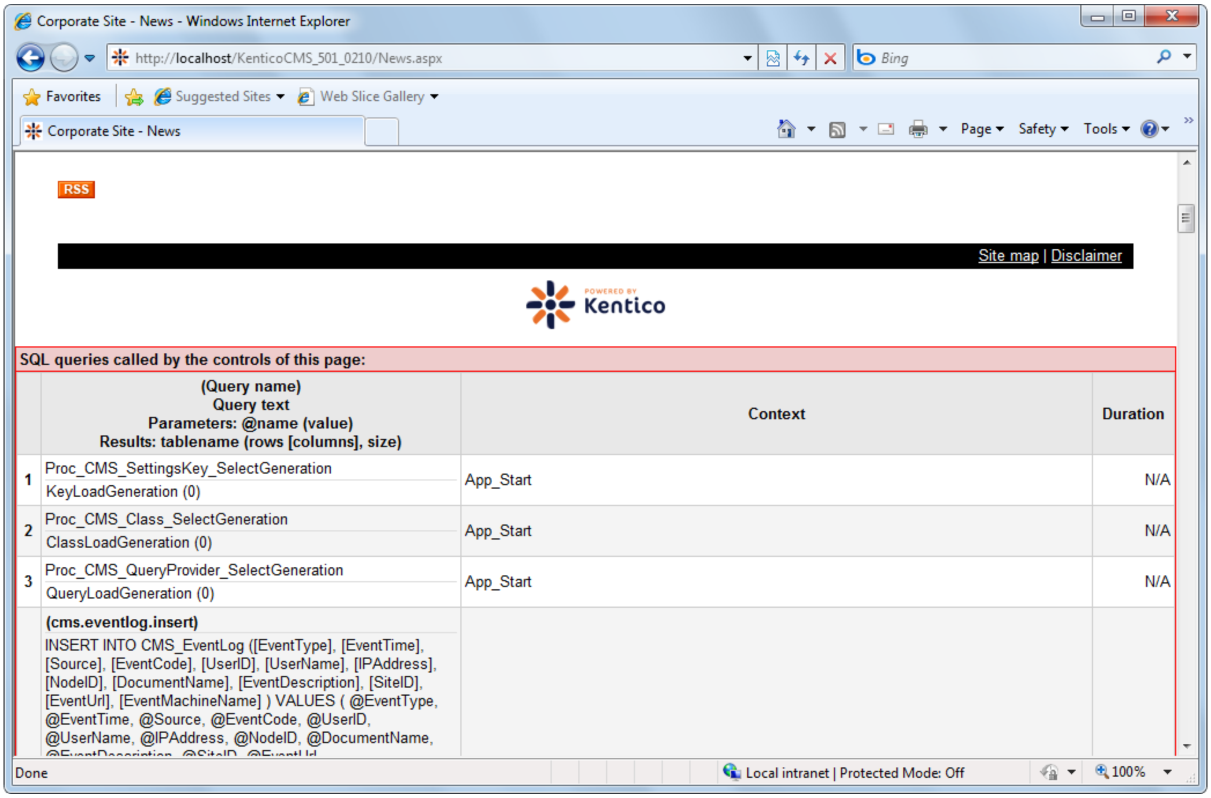
- Cache access
- SQL queries
- Page ViewState
- Output
- Security
- Requests
- Worker threads

You can enable all these debugs at once using the bulk keys.



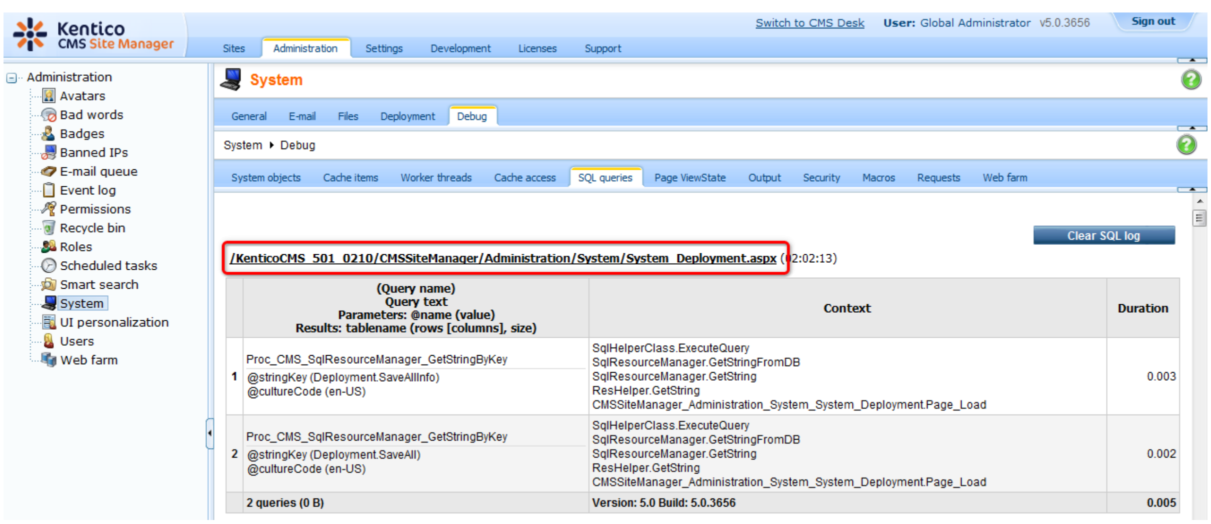
Debugging on the live site

You can also enable debugging directly on the live site. In this case, the debug log is displayed at the bottom of each loaded page. Like UI debugs, live site debugging too can be enabled either separately using the dedicated keys (follow the links above) or all at once using the bulk keys.

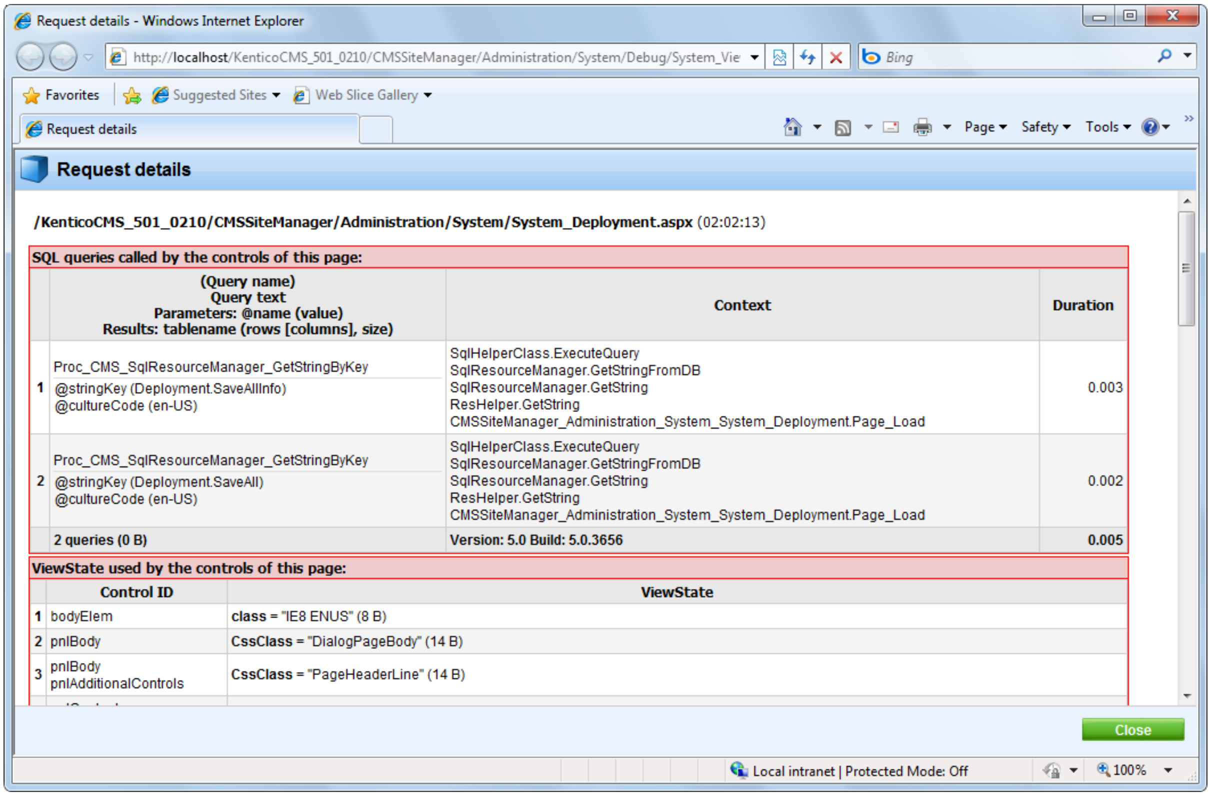


Request details

You can display request details for each debugged request by clicking its URL in any debug log either in the UI or on the live site.



After doing so, a new pop-up window gets opened, listing the request's debug log for all enabled debugs.



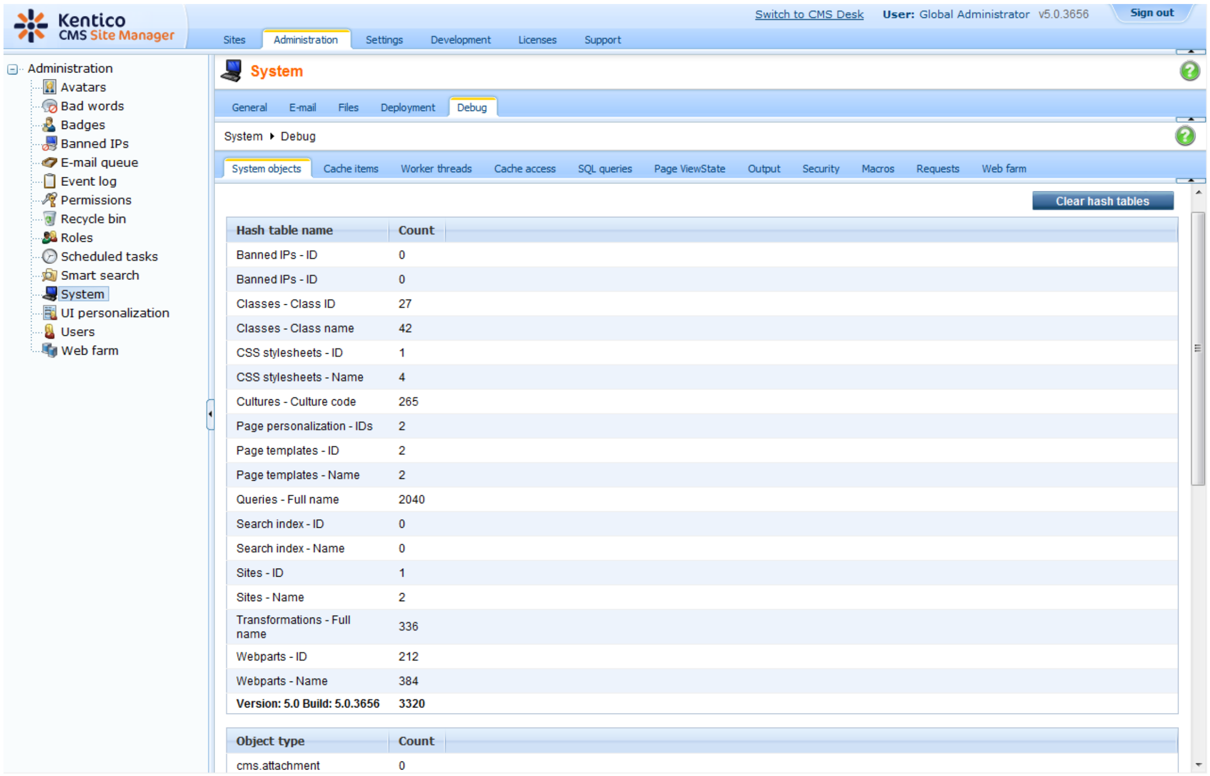
7.5.2 Particular debugs

7.5.2.1 System objects debug

On the **Debug -> System objects** tab, you can see the numbers of objects currently stored in hash tables on the server.

The first table shows the names of particular **hash tables** and the numbers of records in them. The second table shows particular **object types** and the numbers of currently hashed objects of each type.

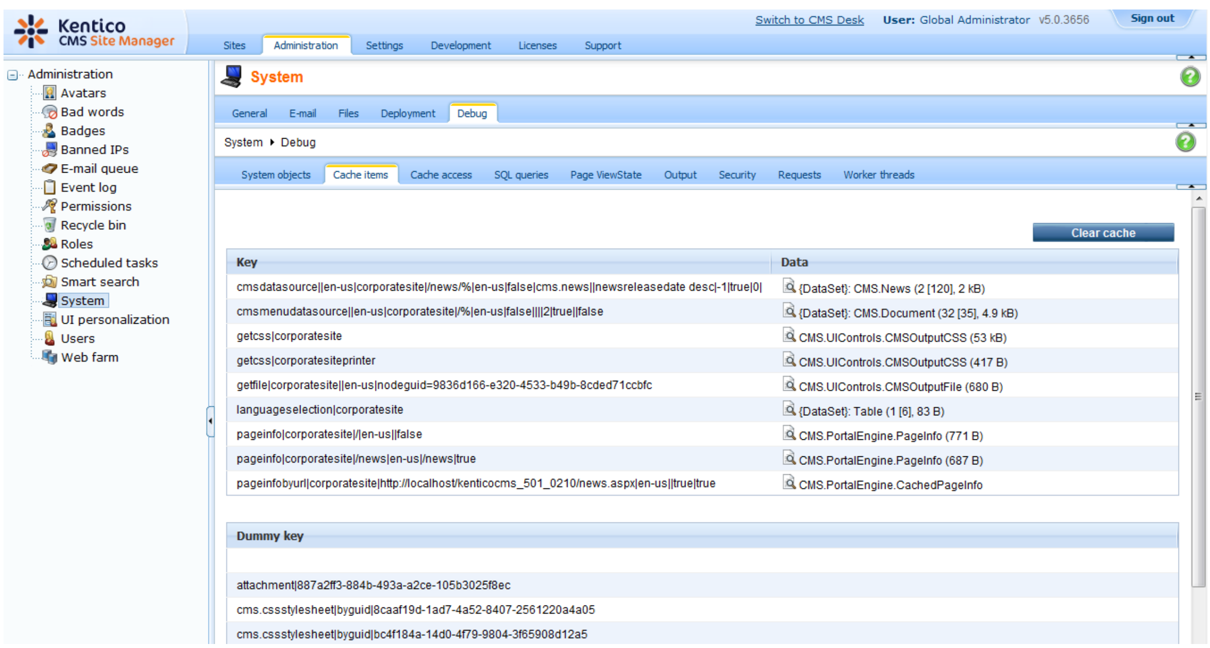
All hash tables can be cleared using the **Clear hash tables** button.



7.5.2.2 Cache items debug

On the **Debug -> Cache items** tab, you can see which **Keys** and **Dummy keys** are currently stored in the system cache. For more information about caching in Kentico CMS, please refer to this topic.

All dummy keys can be cleared from the cache using the **Clear cache** button.



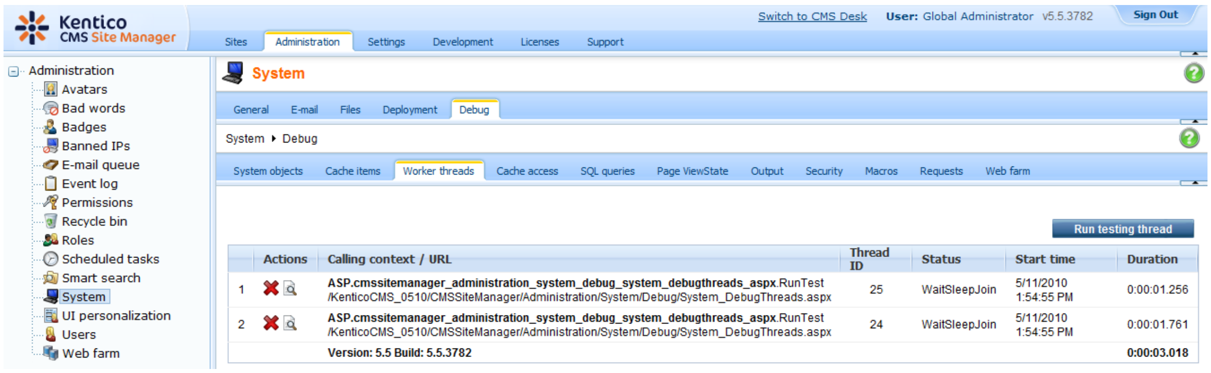
7.5.2.3 Worker threads debug

On the **Debug -> Worker threads** tab, you can see which worker threads are currently running in the system. This is particularly useful if your application is consuming significantly more resources than it normally should. It may often be caused by threads running in the background, e.g. threads for Smart Search re-indexing or Mass e-mail sending.

You can perform the following actions with the listed threads:

- **Cancel** - cancels execution of the thread (useful if the original window where the thread was started is already closed)
- **View** - displays a pop-up window showing the current content of the thread log (in case that there is a log)

The **Run testing thread** button can be used to launch a testing thread in order to verify that threads debugging works correctly.



7.5.2.4 Cache access debug

Cache access debugging can be turned on by adding one or more of the following keys into the *AppSettings* section of your *web.config* file:

- **CMSDebugCache** - enables the Cache access tab in Administration -> System -> Debug
- **CMSDebugCacheLive** - enables cache access debugging on the live site
- **CMSDebugAllCaches** - enables cache access debugging for all operations (including the UI)
- **CMSDebugCacheLogLength** - maximal length of the cache access debug log (number of preserved records)
- **CMSLogCache** - if applied, cache access debug log will be stored into *~/App_Data/logcache.log*

Here is a list of these keys for easy copy&paste into your *web.config*:

```
<add key="CMSDebugCache" value="true" />
<add key="CMSDebugCacheLive" value="true" />
<add key="CMSDebugAllCaches" value="true" />
<add key="CMSDebugCacheLogLength" value="10" />
<add key="CMSLogCache" value="true" />
```

Cache access debugging can also be enabled using the bulk keys.

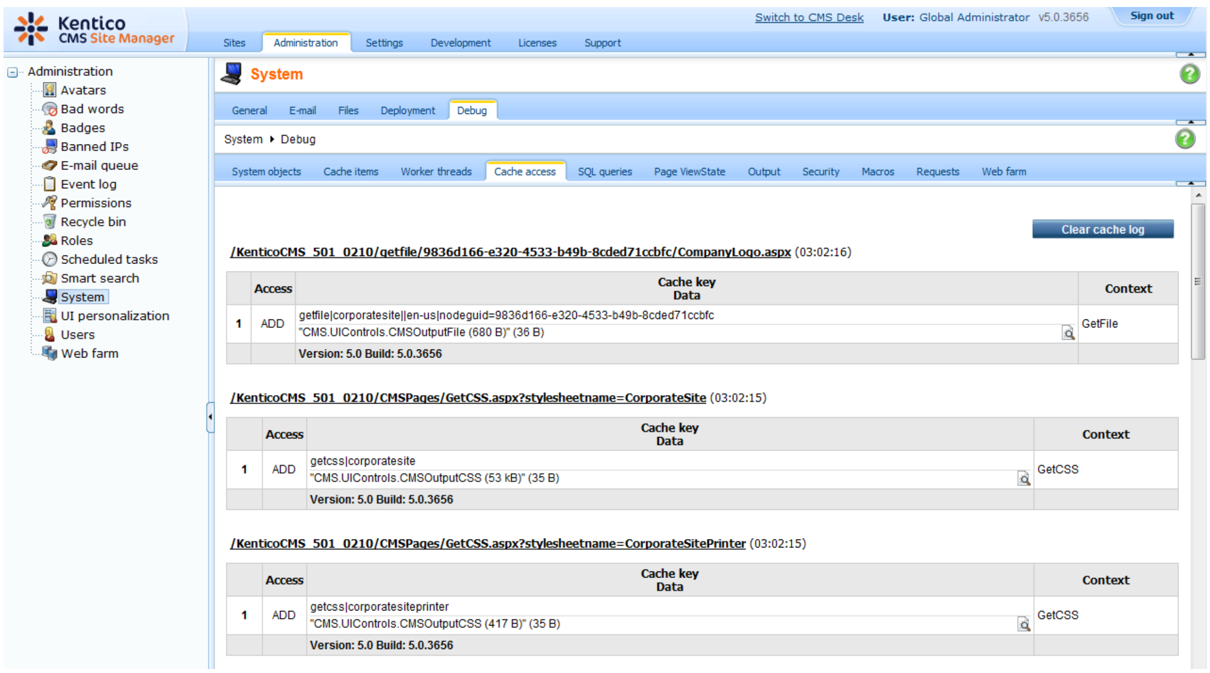
User interface

On the **Debug -> Cache access** tab, you can see a log of accesses to the system cache.

For each access, you can see the URL of the page on whose loading the cache was accessed. The table below the URL always shows the type of **Access**, the accessed **Cache key**, the object type and size of cached **Data** and the **Context** from which the cache was accessed. For more information about caching in Kentico CMS, please refer to this topic.

All dummy keys can be cleared from the cache using the **Clear cache** button.

By clicking the **View object** ( ) icon, you can display details of the cached object in a new window. Clicking the **Clear cache log** button clears all records in this debug log.



7.5.2.5 SQL queries debug

SQL queries debugging can be turned on by adding one or more of the following keys to *AppSettings* section of your project's *web.config* file:

- **CMSDebugSQLQueries** - enables the SQL queries tab in Site Manager -> Administration -> System -> Debug
- **CMSDebugSQLQueriesLive** - enables SQL queries debugging on the live site
- **CMSDebugAllSQLQueries** - enables debugging of all SQL queries (including the UI)
- **CMSDebugSQLQueriesLogLength** - maximal length of the SQL queries debug log (number of preserved records)
- **CMSLogSQLQueries** - if applied, SQL queries debug log will be saved into *~/App_Data/logsql.log*

Here is a list of these keys for easy copy&paste into your *web.config*:

```
<add key="CMSDebugSQLQueries" value="true" />
<add key="CMSDebugSQLQueriesLive" value="true" />
<add key="CMSDebugAllSQLQueries" value="true" />
<add key="CMSDebugSQLQueriesLogLength" value="10" />
<add key="CMSLogSQLQueries" value="true" />
```

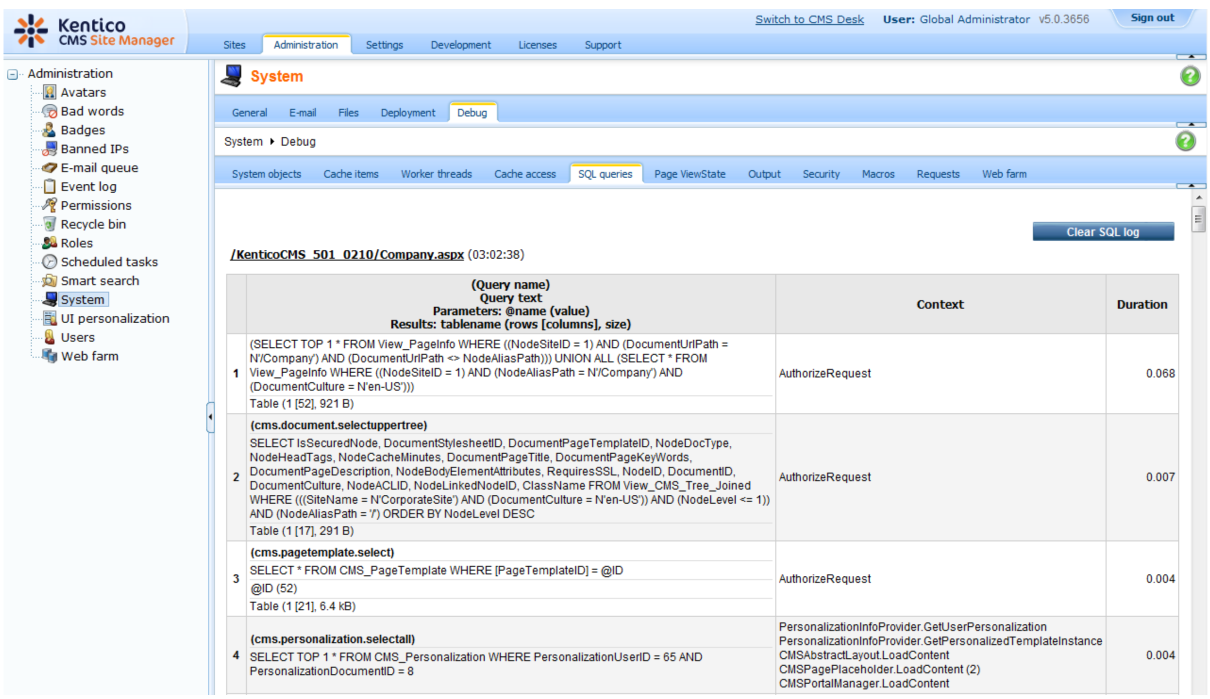
SQL queries debugging can also be enabled using the bulk keys.

User interface

On the **Debug -> SQL queries** tab, you can see a log of SQL queries called on loading of particular pages. For each record, there is row with a URL of the loaded page followed by its time of display. Below it, you can find a table listing all SQL queries executed when loading the page.

For each query, there are two lines in the table. The first line contains the exact text of the query, while the second line contains the number of loaded rows, columns in each row and the size of the loaded data. The Context column shows where the query was called from. The last column of the table displays the exact duration of query execution. The last line of the table displays the overall page loading time.

Clicking the **Clear SQL log** button clears all records in this debug log.



7.5.2.6 Page ViewState debug

Page ViewState debugging can be turned on by adding one or more of the following keys into the *AppSettings* section of your *web.config* file:

- **CMSDebugViewState** - enables the Page ViewState tab in Administration -> System -> Debug
- **CMSDebugViewStateLive** - enables ViewState debugging on the live site

- **CMSDebugAllViewStates** - enables ViewState debugging for all operations (including the UI)
- **CMSDebugViewStateLogLength** - maximal length of the ViewState debug log (number of preserved records)

Here is a list of these keys for easy copy&paste into your *web.config*:

```
<add key="CMSDebugViewState" value="true" />
<add key="CMSDebugViewStateLive" value="true" />
<add key="CMSDebugAllViewStates" value="true" />
<add key="CMSDebugViewStateLogLength" value="10" />
```

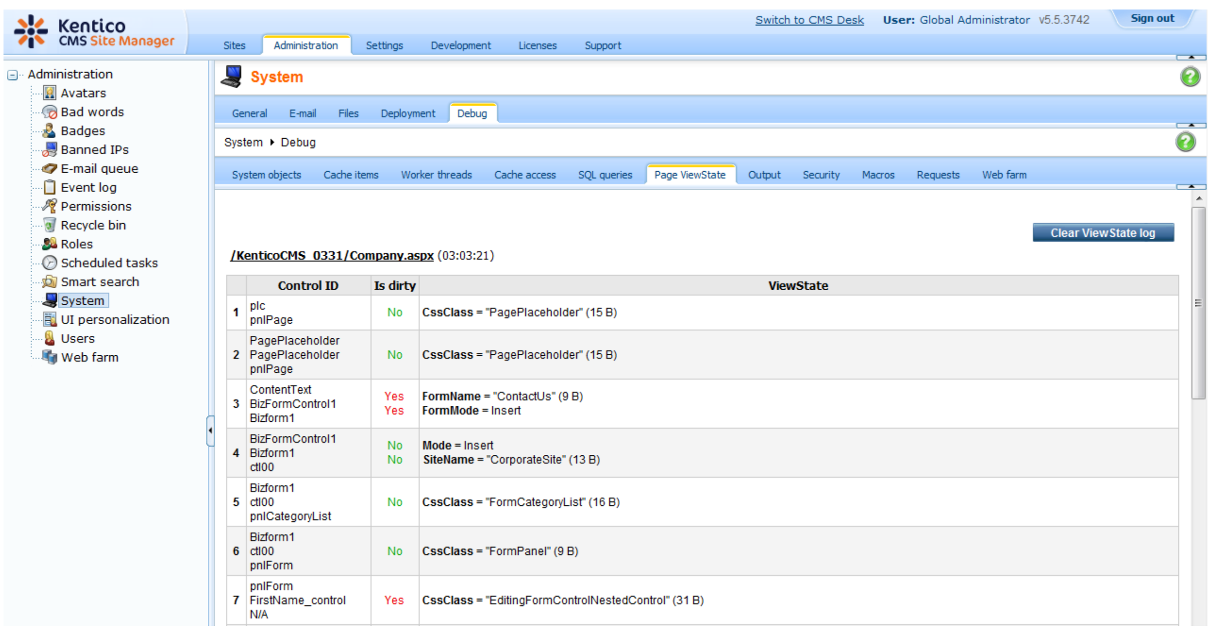
Page ViewState debugging can also be enabled using the bulk keys.

User interface

On the **Debug -> Page ViewState** tab, you can see the ViewState of particular controls on recently loaded pages. The **Is dirty** column indicates if the item was added to ViewState after call of the *TrackViewState()* method (happens typically in *OnInit*).

The log can be cleared using the **Clear ViewState log** button.

Please note: ViewState of the controls is retrieved using reflection. Therefore this debug may not work properly under medium trust environment or other specific circumstances.



7.5.2.7 Output debug

Output debugging can be turned on by adding one or more of the following keys into the *AppSettings* section of your *web.config* file:

- **CMSDebugOutput** - enables the Output tab in Site Manager -> Administration -> System -> Debug

- **CMSDebugOutputLive** - enables output debugging on the live site
- **CMSDebugAllOutputs** - enables output debugging for all operations (including the UI)
- **CMSDebugOutputLogLength** - maximal size of the output debug log (number of preserved records)
- **CMSLogOutputToFile** - if applied, output debug log is saved into *~/App_Data/logoutput.log*

Here is a list of these keys for easy copy&paste into your *web.config*:

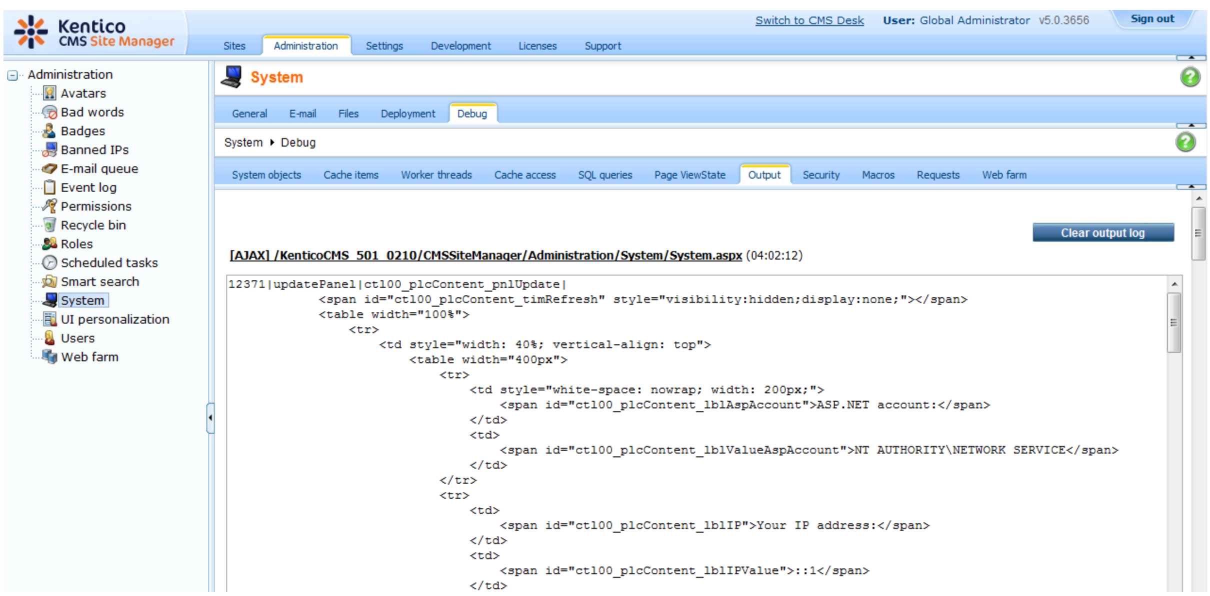
```
<add key="CMSDebugOutput" value="true" />
<add key="CMSDebugOutputLive" value="true" />
<add key="CMSDebugAllOutputs" value="true" />
<add key="CMSDebugOutputLogLength" value="10" />
<add key="CMSLogOutputToFile" value="true" />
```

Output debugging can also be enabled using the bulk keys.

User interface

On the **Debug -> Output** tab, you can see the exact output code of recently displayed pages. This is particularly useful in case of AJAX request, whose code can not be viewed as part of the page source directly in the browser.

The log can be cleared using the **Clear output log** button.



7.5.2.8 Security debug

Security debugging can be turned on by adding one or more of the following keys into the *AppSettings* section of your *web.config* file:

- **CMSDebugSecurity** - enables the Security tab in Site Manager -> Administration -> System -> Debug
- **CMSDebugSecurityLive** - enables security debugging on the live site
- **CMSDebugAllSecurity** - enables security debugging for all operations (including the UI)

- **CMSDebugSecurityLogLength -** maximal length of the security debug log (number of preserved records)
- **CMSLogSecurity** - if applied, output debug log will be saved into *~/App_Data/logsecurity.log*

Here is a list of these keys for easy copy&paste into your *web.config*:

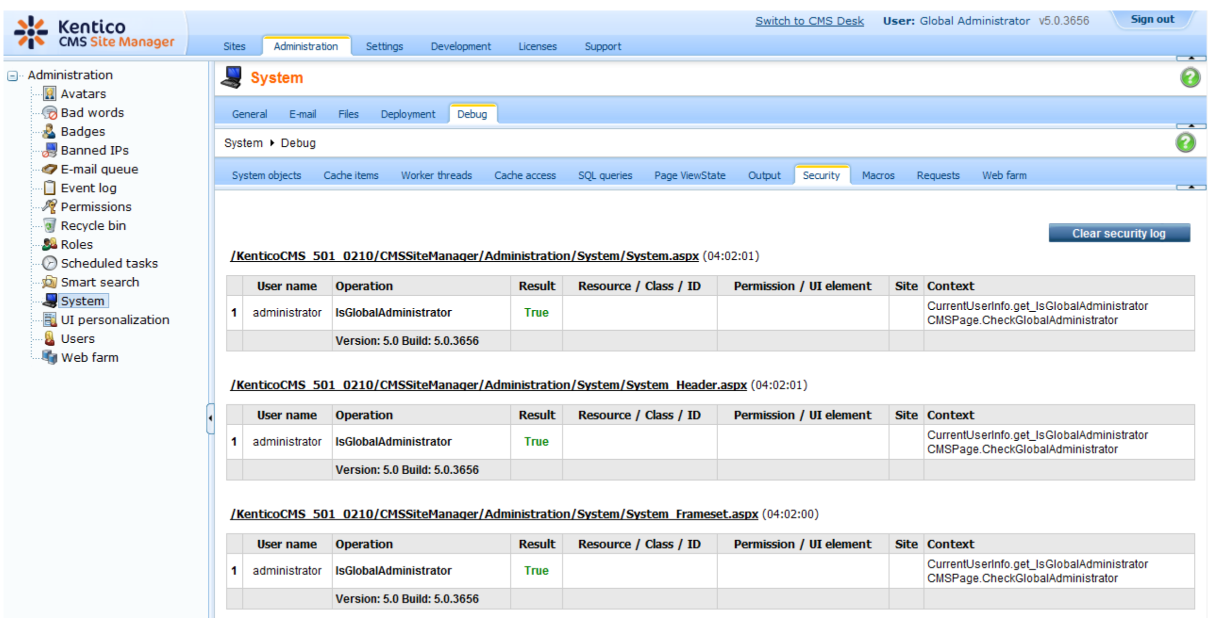
```
<add key="CMSDebugSecurity" value="true" />
<add key="CMSDebugSecurityLive" value="true" />
<add key="CMSDebugAllSecurity" value="true" />
<add key="CMSDebugSecurityLogLength" value="10" />
<add key="CMSLogSecurity" value="true" />
```

Security debugging can also be enabled using the bulk keys.

User interface

On the **Debug -> Security** tab, you can see which security checks were recently performed on the site. This is particularly useful if you want to quickly find out why some user is not able to access some section of the UI or gets the *Access denied* page displayed.

The log can be cleared using the **Clear security log** button.



7.5.2.9 Macros debugging

Macros debugging can be turned on by adding one or more of the following keys into the *AppSettings* section of your *web.config* file:

- **CMSDebugMacros** - enables the Macros tab in Site Manager -> Administration -> System -> Debug
- **CMSDebugMacrosLive** - enables macros debugging on the live site
- **CMSDebugAllMacros** - enables macros debugging for all operations (including the UI)
- **CMSDebugMacrosLogLength** - maximal length of the macros debug log (number of preserved

records)

Here is a list of these keys for easy copy&paste into your *web.config*:

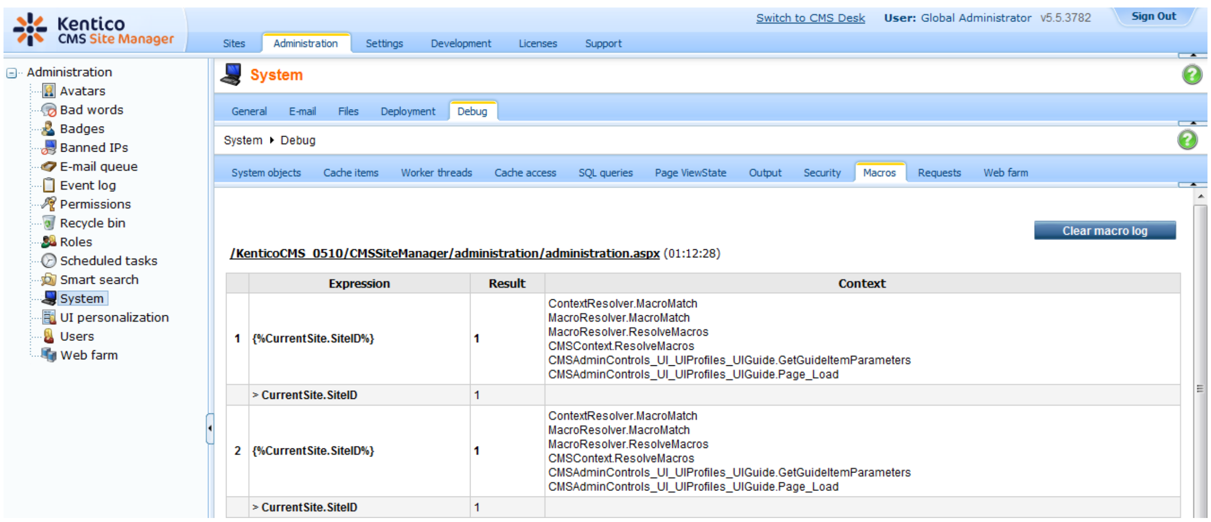
```
<add key="CMSDebugMacros" value="true" />
<add key="CMSDebugMacrosLive" value="true" />
<add key="CMSDebugAllMacros" value="true" />
<add key="CMSDebugMacrosLogLength" value="20" />
```

Macros debugging can also be enabled using the bulk keys.

User interface

On the **Debug -> Macros** tab, you can see which macros were recently resolved. For each macro, its exact expression, result (1 - resolved successfully, 0 - not resolved) and context from which it was called is logged. This is useful if you want to analyze how macros were processed.

The log can be cleared using the **Clear macro log** button.



7.5.2.10 Request debug

Request debugging can be turned on by adding one or more of the following keys into the *AppSettings* section of your *web.config* file:

- **CMSDebugRequests** - enables the Request tab in Site Manager -> Administration -> System -> Debug
- **CMSDebugRequestsLive** - enables request debugging on the live site
- **CMSDebugAllRequests** - enables request debugging for all pages (including the UI)
- **CMSDebugRequestsLogLength** - if applied, output debug log will be saved into *~/App_Data/logrequests.log*

Here is a list of these keys for easy copy&paste into your *web.config*:

```
<add key="CMSDebugRequests" value="true" />
<add key="CMSDebugRequestsLive" value="true" />
<add key="CMSDebugAllRequests" value="true" />
<add key="CMSDebugRequestsLogLength" value="10" />
```

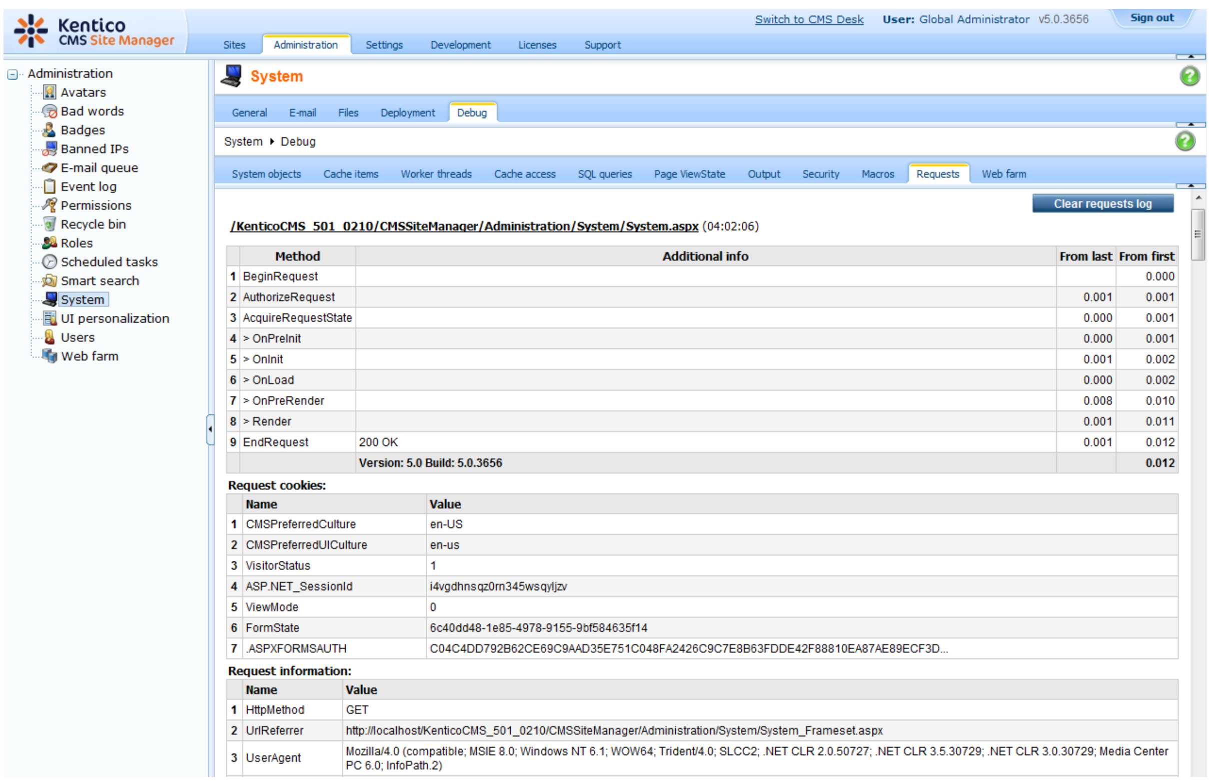
Request debugging can also be enabled using the bulk keys.

User interface

On the **Debug -> Requests** tab, you can see the time each of the recent page requests took to process. You not only see the overall time, but also separate times of particular parts of each requests, along with other detailed information about the request.

This is particularly useful if your response time is too large and you need to figure out why. You can basically see whether the issue is outside the application or inside it by comparing the real response time and response time spent in the application, what data came with the request and from which context.

The log can be cleared using the **Clear request log** button.



7.5.2.11 Web farm debugging

Web farm debugging can be turned on by adding one or more of the following keys into the *AppSettings* section of your *web.config* file:

- **CMSDebugWebFarm** - enables the Web farm tab in Site Manager -> Administration -> System ->

Debug

- **CMSDebugAllWebFarm** - enables web farm debugging for all operations (including the UI)
- **CMSDebugWebFarmLogLength** - maximal length of the web farm debug log (number of preserved records)

Here is a list of these keys for easy copy&paste into your *web.config*:

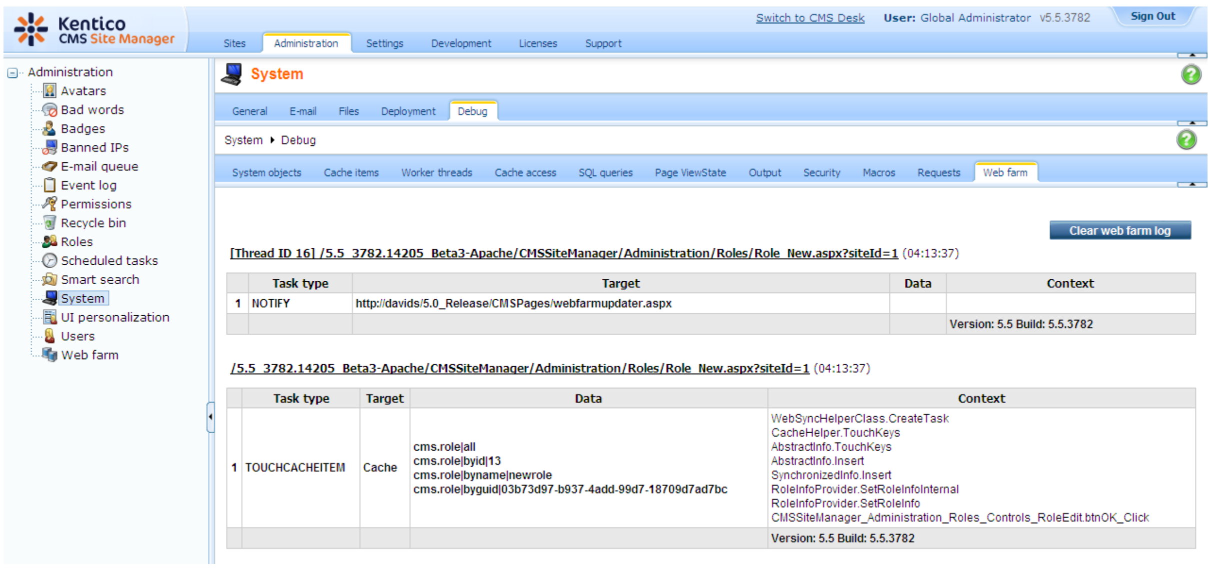
```
<add key="CMSDebugWebFarm" value="true" />
<add key="CMSDebugAllWebFarm" value="true" />
<add key="CMSDebugWebFarmLogLength" value="20" />
```

Web farm debugging can also be enabled using the bulk keys.

User interface

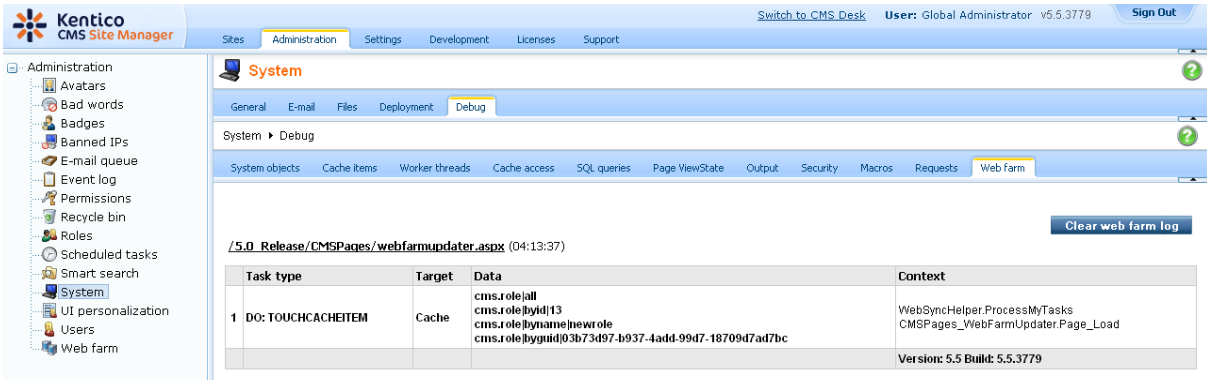
On the **Debug -> Web farm** tab, you can see which web farm tasks were logged and which servers were notified about the changes. This is particularly useful if you want to find out if the web farm works effectively and synchronizes correct data, or generally if you want to troubleshoot web-farm-related issues.

The following screenshot shows how the debug looks on the source server (the server where the data was modified). The bottom part show the logged tasks, while the top part shows asynchronous notifications of other web farm servers.



The second screenshot shows how the debug looks on the target server after it is notified about the modification. You can recognize the task by the *DO:* prefix, which indicates that the synchronization task was (is being) processed.

The log can be cleared using the **Clear web farm log** button.



7.5.3 Bulk keys for all debugs

You can also add one of the following bulk keys to the *AppSettings* section of your *web.config* file in order to enable all the debugs with a single key.

- **CMSDebugEverything** - enables all debugging sub-tabs in Site Manager -> Administration -> System -> Debug
- **CMSDebugAllForEverything** - enables all debugging sub-tabs, while UI operations are debugged too
- **CMSDebugEverythingLive** - enables all debugging on the live site (the debug log will be displayed at the bottom of the page)
- **CMSDebugEverythingLogLength** - limits the maximal length of the debug log (number of preserved log records)
- **CMSLogEverythingToFile** - logs all possible operations (including Event log and E-mail sending log) into *.log* files stored in *~/App_Data/*

Here is a list of these keys for easy copy&paste into your *web.config*:

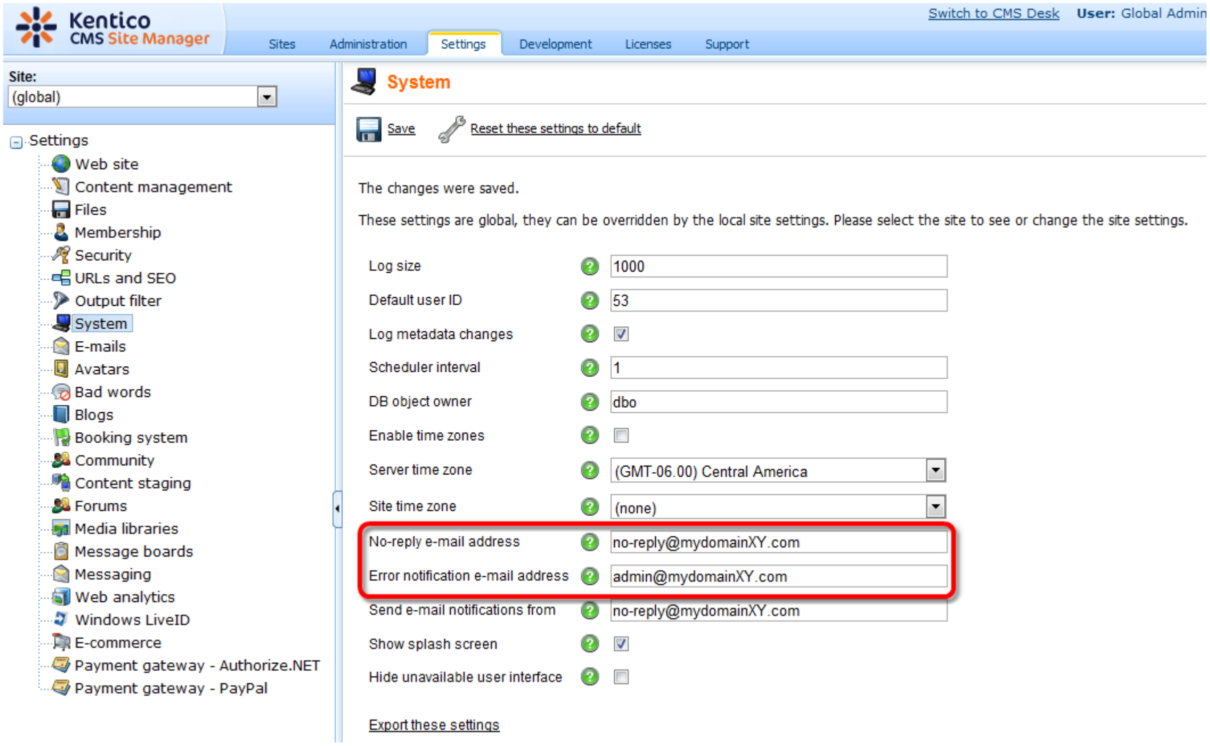
```
<add key="CMSDebugEverything" value="true"/>
<add key="CMSDebugAllForEverything" value="true"/>
<add key="CMSDebugEverythingLive" value="true"/>
<add key="CMSDebugEverythingLogLength" value="20"/>
<add key="CMSLogEverythingToFile" value="true"/>
```

These keys enable all the available debugs listed on this page, while the dedicated keys for particular debugs have higher priorities. This means that you can for example enable all debugs using *CMSDebugEveryting* with the value set to *true* and hide one particular sub-tab by using its dedicated key with the value set to *false* at the same time.

7.5.4 System error notifications

If you go to **Site Manager -> Settings -> System** and fill in an e-mail address into the **Error notification e-mail address**, notifications about internal errors in Kentico CMS system will be sent to this address whenever such an error occurs.

The e-mail address specified in the **No-reply e-mail address** field will be used as the sender ('From') e-mail address.



7.6 Document types and transformations

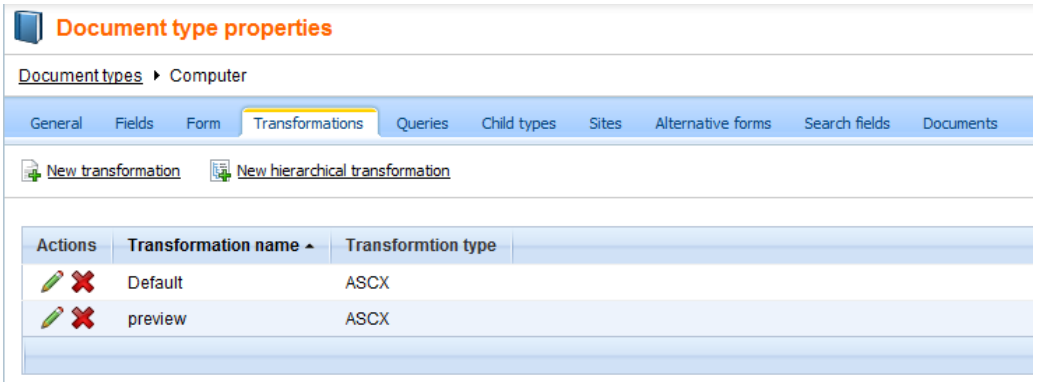
7.6.1 Overview

Each document in Kentico CMS is of some type. Each document type has its own:

- fields (data structure)
- editing form layout
- transformations (design)
- queries

and other settings.

Document types are fully customizable - you can add, modify and delete custom fields. The advantage of using custom document types is that you can define custom structure of documents and store content (data) separated from design. This can be done in **Site Manager -> Development -> Document types**.



General properties of a document type

Document type display name	The name of the site displayed to the users.
Document type code name	The name of the site used in the code.
New page	URL of the page that will be used for creating new documents of this type.
Editing page	URL of the editing page that will be used when the document is displayed in the editing mode, using the Page tab.
Editing form	URL of the editing page that will be used when the document is displayed in the editing mode, using the Form tab.
Preview page	URL of the editing page that will be used when the document is displayed in the preview mode.
List page	URL of the editing page that will be used when the document is displayed in the list mode.
Use publish from/publish to	Indicates if publish from/to fields should be displayed for this document type.
Show template selection	Indicates if users must select some page template first when they create a new document of this type.
Default page template	The page template used by default when the document is created. If no page template is specified, the document inherits the parent page template.
Behaves as Page (menu item) type	Indicates if the document type has similar behavior as Page (menu item) document type. The default view mode for such a document type is the Page tab. Viewer web parts are automatically set to display the child documents if the path is not configured and the document does not inherit its parent template by default.
Document is product type	Indicates if documents of this type are products of the E-commerce module.

Transformations are pieces of code that determine how Kentico CMS documents, or certain parts of them, are rendered by listing web parts and controls. They take raw data from the Kentico CMS database and *transform* it into the form you wish it to appear in. This makes them a crucial tool when displaying documents and document related data on the pages of your website. Transformations can be accessed in **Site Manager -> Development -> Document types -> Edit ( ) <document_type>**, on the **Transformations** tab.

- If you would like to learn about the functionality of transformations and/or would like to see examples of the code, please refer to the Transformations topic.
- If you would like to learn how to define a new document type, please refer to the Defining a new document type topic.
- If you would like to see an example of writing transformations for a particular document type, please refer to the Writing transformations topic.
- If you would like to learn how to add your own functions to transformations, please refer to the Adding custom functions to transformations topic.
- To learn how to add context menus to web parts displaying **Users** or **Groups**, please refer to the Context menus in transformations topic.

Kentico CMS E-commerce Guide contains several examples of the use of document types and transformations. See e.g. Integration with your existing Kentico CMS web site -> Web parts -> Random products, Customizing the product fields -> Defining a new product type or Customizing product design in the same section of the E-commerce Guide. Please note that these practical examples do not concern the whole functionality of the Document types and transformations section but focus on its specific application.

You will need to have the E-commerce Site sample website installed to follow Kentico CMS E-commerce Guide.

A number of practical examples of the use of document types and transformations are available in Kentico CMS Community Site Guide; please note that these examples are again specific applications within the context of the Community Site sample website. See e.g. Part 2 -> Creating the Blogs section -> Creating the Blogs page in Kentico CMS Community Site Guide.

Please note that Kentico CMS Personal Site Guide also contains several topics where the use of document types is mentioned.

7.6.2 Transformations

The functionality of transformations is very similar to that of templates used by standard ASP.NET list controls such as the Repeater. The main difference is that our transformations are stored separately in the database and can easily be used repeatedly. They can be assigned to listing web parts or controls through the appropriate **Transformation** properties.

The code of transformations depends on their type. ASCX transformations are the most common, they can contain a mix of HTML elements, embedded controls, standard ASP.NET data binding expressions and methods, such as *Eval()*, and special methods designed to be used in transformations. XSLT type transformations need to be in valid XML format and can contain standard XSL elements.

The use of transformations is supported by all web parts that can display document data, as well as by those listing controls that are designed to work directly with Kentico CMS documents.

Transformations are categorized under the document types that they are supposed to display. They can be managed in the Kentico CMS administration interface at **Site Manager -> Development -> Document types -> ... Edit document type ... -> Transformations**. Some document types do not represent an object but serve only as a container for transformations and queries.

The full name used to identify transformations uses the following format: *<document type code name>.<transformation name>*

The sample sites include many transformations for all document types and you can modify them or write new transformations to suit any of your requirements.

Example

The code of the **Ecommerce.Transformations.Product_SimplePreview** ASCX transformation, which is used to display key information about products, looks like this:

```
<div class="ProductPreview">
  <div class="ProductBox">
    <div class="ProductImage">
        <%# EcommerceFunctions.GetProductImage(Eval("SKUImagePath"), 140, Eval(
"SKUName")) %>
    </div>
    <div class="ProductTitle">
        <%# HTMLEncode(ResHelper.LocalizeString(Convert.ToString(Eval("SKUName"
)))) %>
    </div>
    <table class="ProductPrice" cellpadding="0" cellspacing="0">
    <tr>
        <td class="left">Our price: </td>
        <td class="right"><%# EcommerceFunctions.GetFormatedPrice(Eval("SKUPrice"
), Eval("SKUDepartmentID")) %></td>
    </tr>
    </table>
  </div>
</div>
```

When this transformation is assigned to a listing web part or control that has products (SKUs) in its data source, the output code of individual products will contain the values returned by the methods and data binding expressions, like the following example:

```
<div class="ProductPreview">
  <div class="ProductBox">
    <div class="ProductImage">
        <img alt="Samsung SGH E250" src="/KenticoCMS_FilterTest/getmetafile/
da2495b2-ff5f-47cb-b463-4b99d308eadd/CELL_SAMSUNG_SGH_E250.aspx?maxSideSize=140"
border="0" />
    </div>
    <div class="ProductTitle">
        Samsung SGH E250
    </div>
    <table class="ProductPrice" cellpadding="0" cellspacing="0">
      <tr>
        <td class="left">Our price: </td>
        <td class="right">$249.00</td>
      </tr>
    </table>
```

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

The final output of this product on the website will then look like this:

	Please note The CSS stylesheet used by the page or site is applied to the output of the transformation. This example uses the default **Corporate Site** stylesheet.

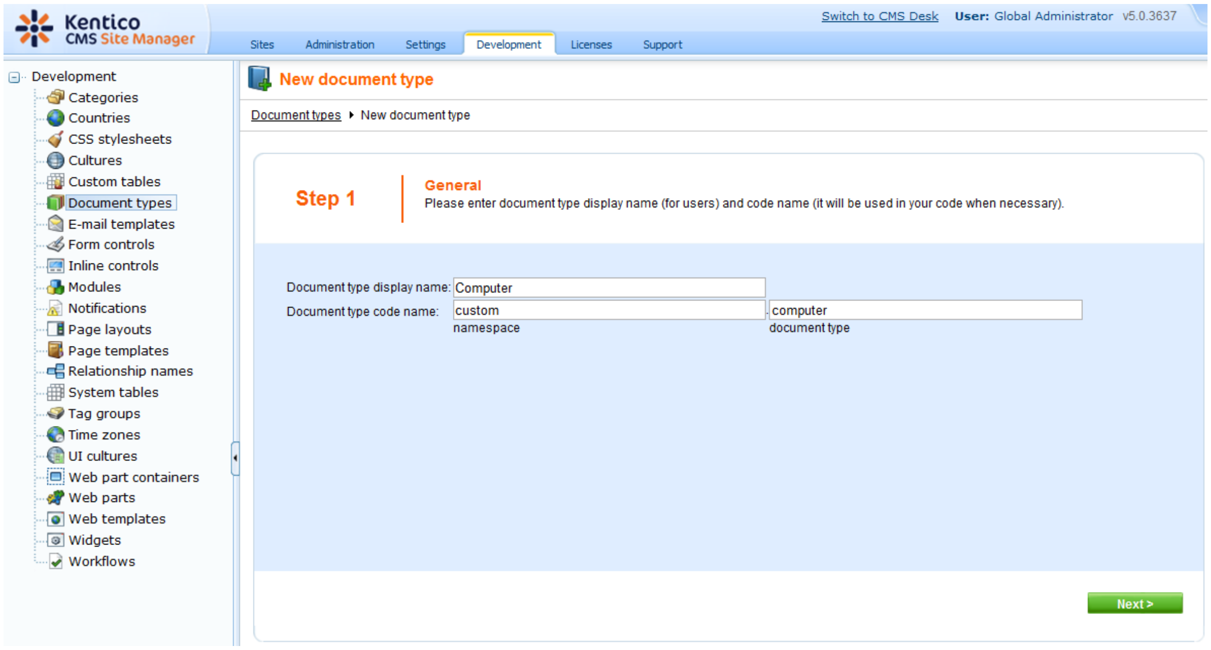
7.6.3 Defining a new document type

In this example, you will learn how to define a new custom document type.

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

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

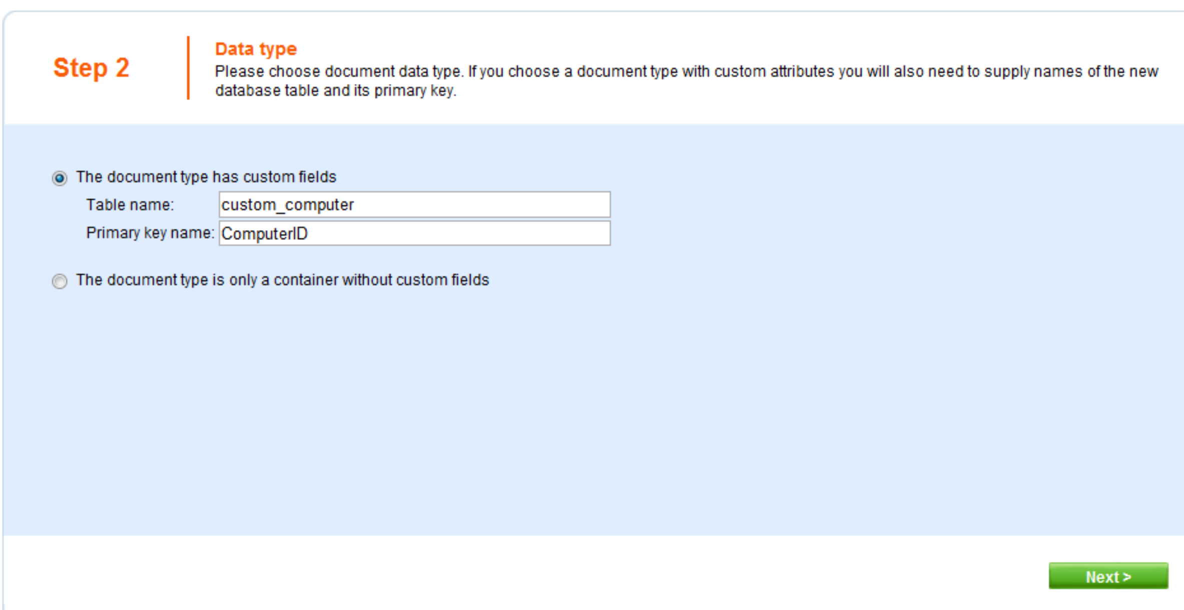
Click **Next**.



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

- **Table name:** CUSTOM_Computer
- **Primary key name:** ComputerID

Click **Next**.



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

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

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

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

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

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

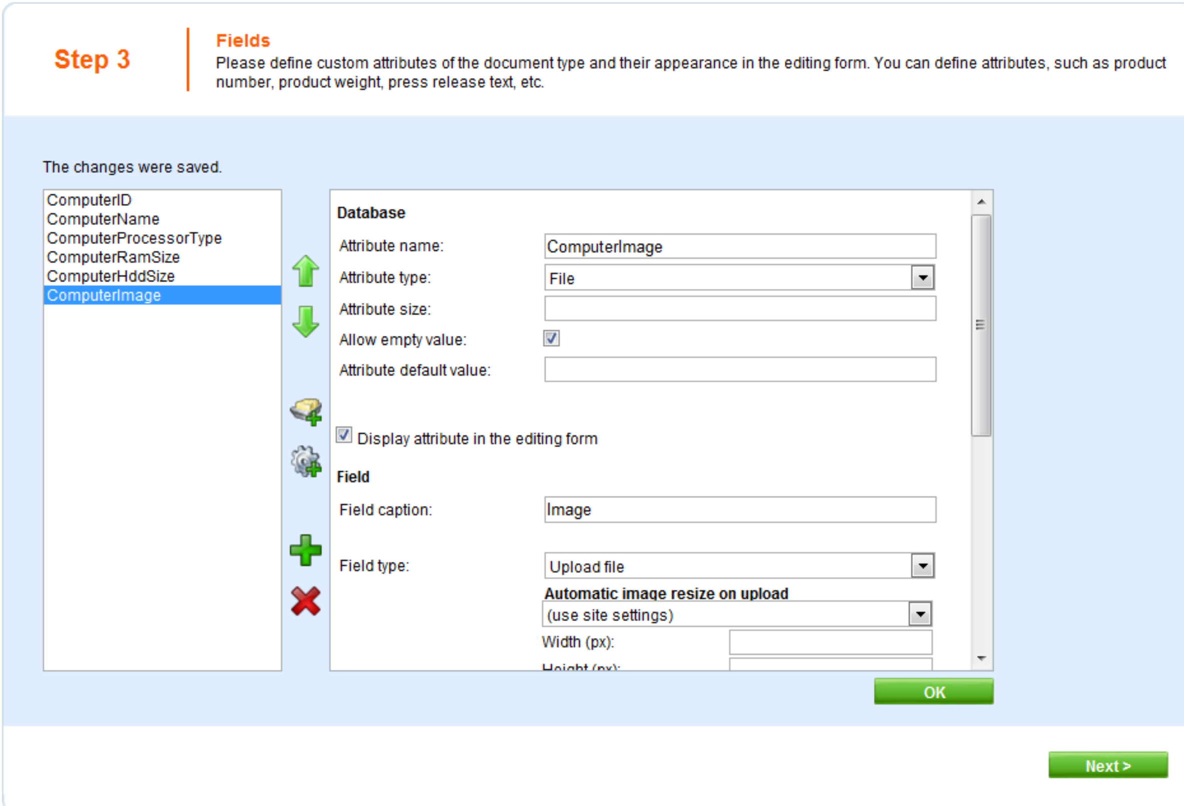
Click **OK**. Click **New attribute** (). Enter:

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

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

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

Click **OK**.

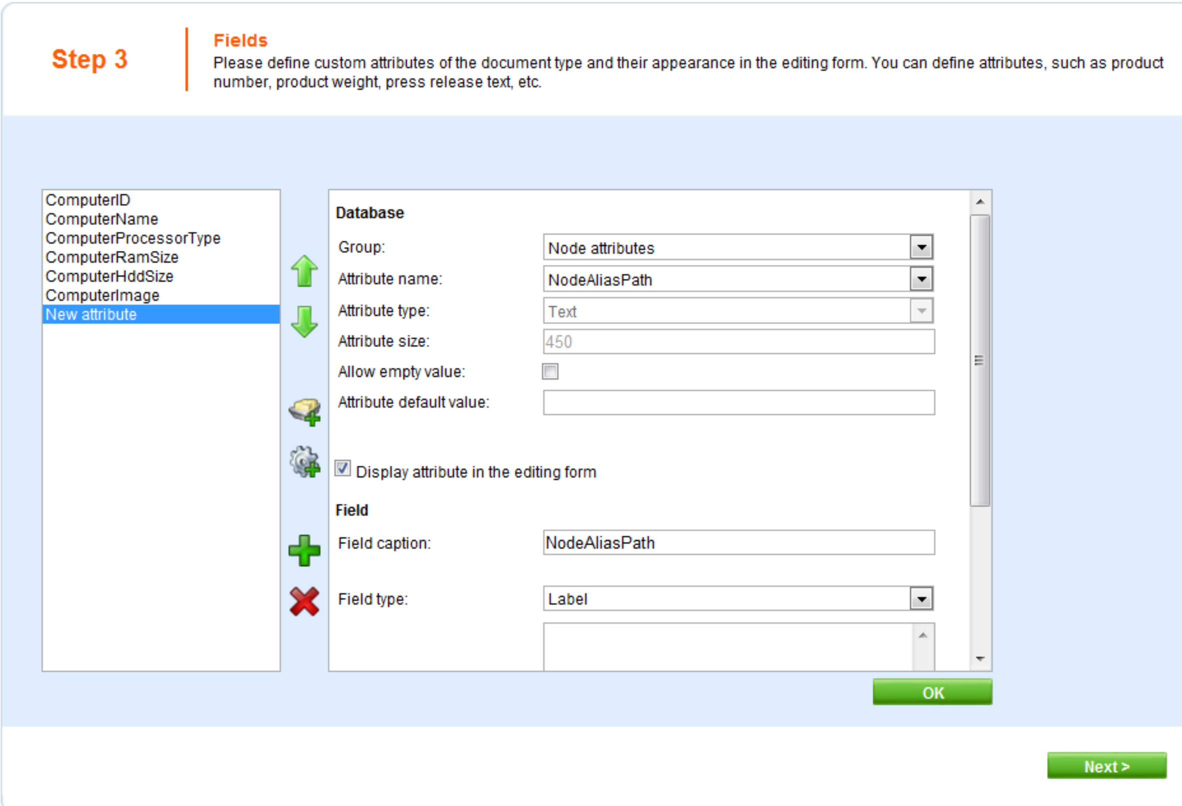


Please note that you can also define system fields to be displayed on the documents' **Form** tab. This can be done by clicking the **Add system attribute** icon. Using the **Group** drop-down list, you can then choose from the following two groups of system fields:

- **Document attribute** - offers the document's system fields
- **Node attribute** - offers the document node's system fields

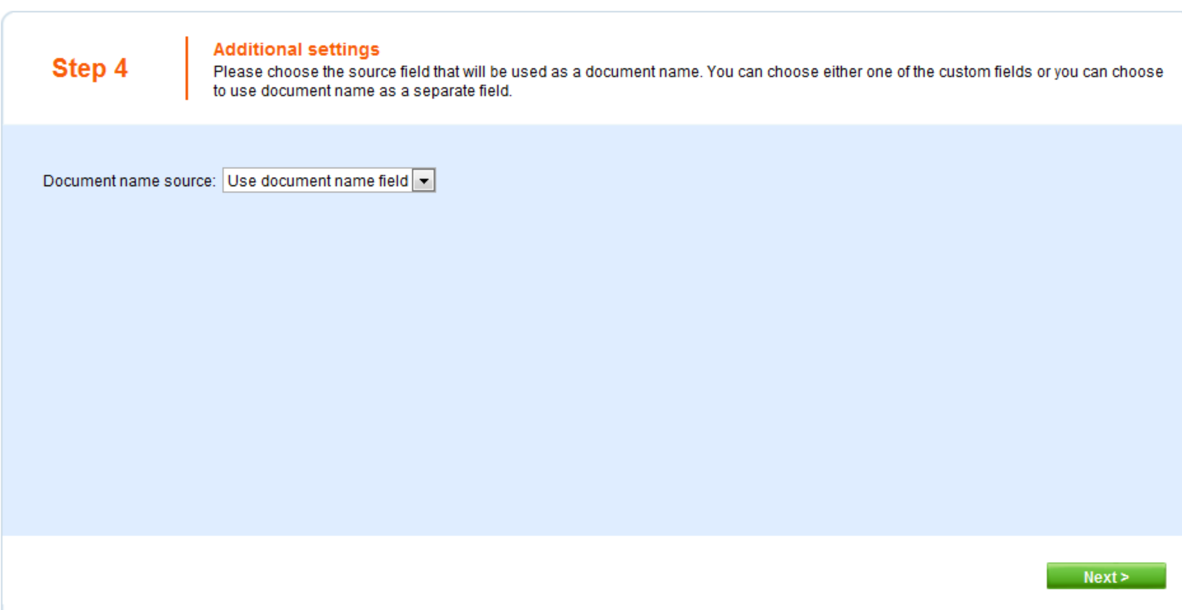
Document or node system fields will then be offered in the **Attribute name** drop-down list. If you leave the **Display attribute in the editing form** check-box checked, the field will be visible on the documents' **Form** tab.

Click **Next**.



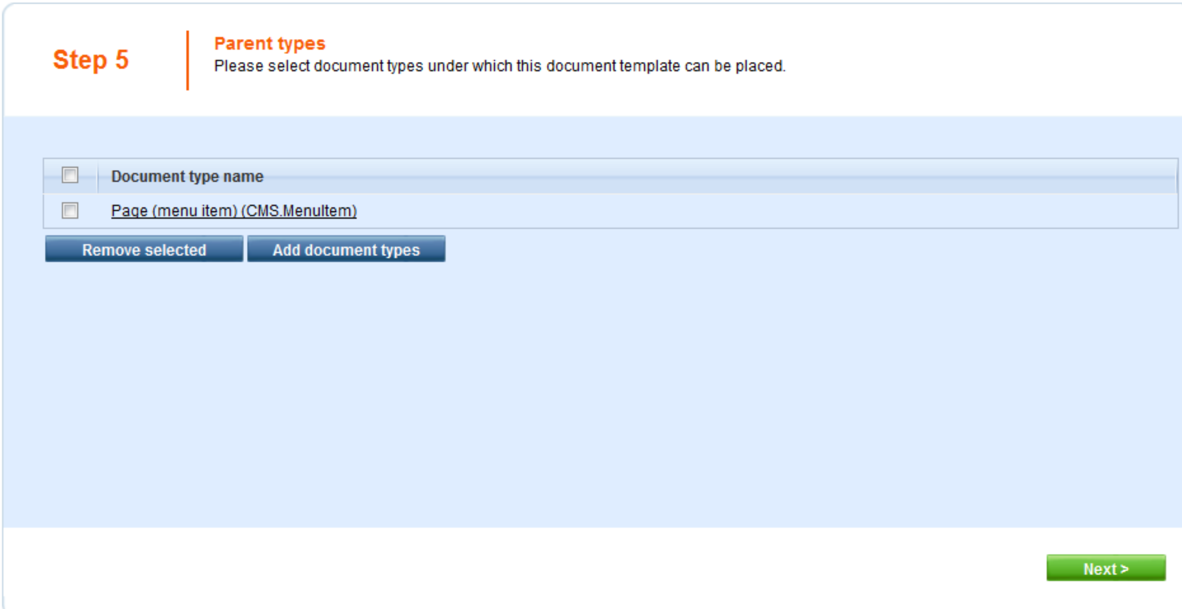
4. Now you need to choose the field that will be used as document name. Choose the **Use document name field** option from the drop down list. It means that when you create a new computer document, its name will be automatically taken from the *ComputerName* value and this value will appear in site navigation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

Click **Next**.

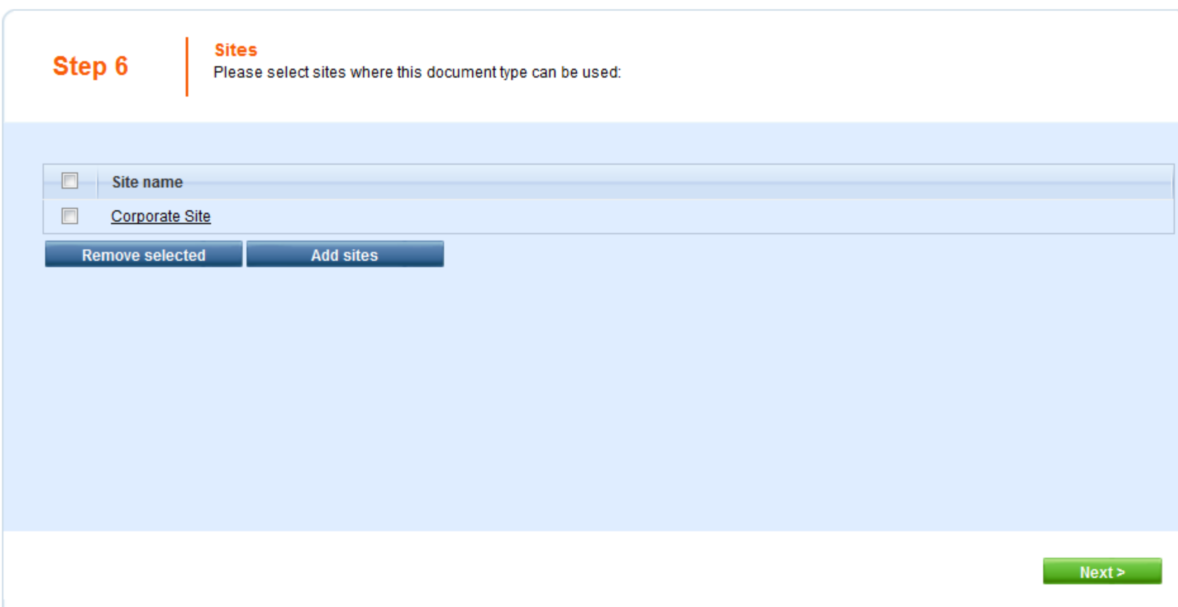


5. In Step 5, you need to select the document types under which computer documents will be displayed. Check only the **Page (menu item)** value, which means that editors will be able to create computer documents only under some page, not under an article or news document in the content tree.

Click **Next**.

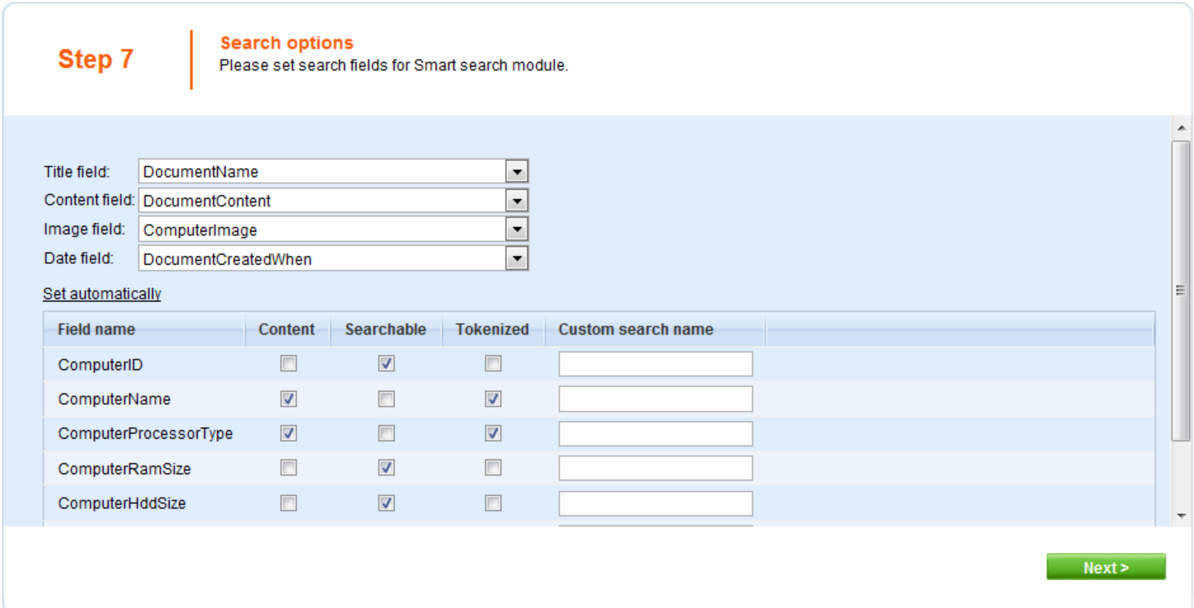


6. In Step 6, you need to choose which websites will use this document type. Check the appropriate website and click **Next**.



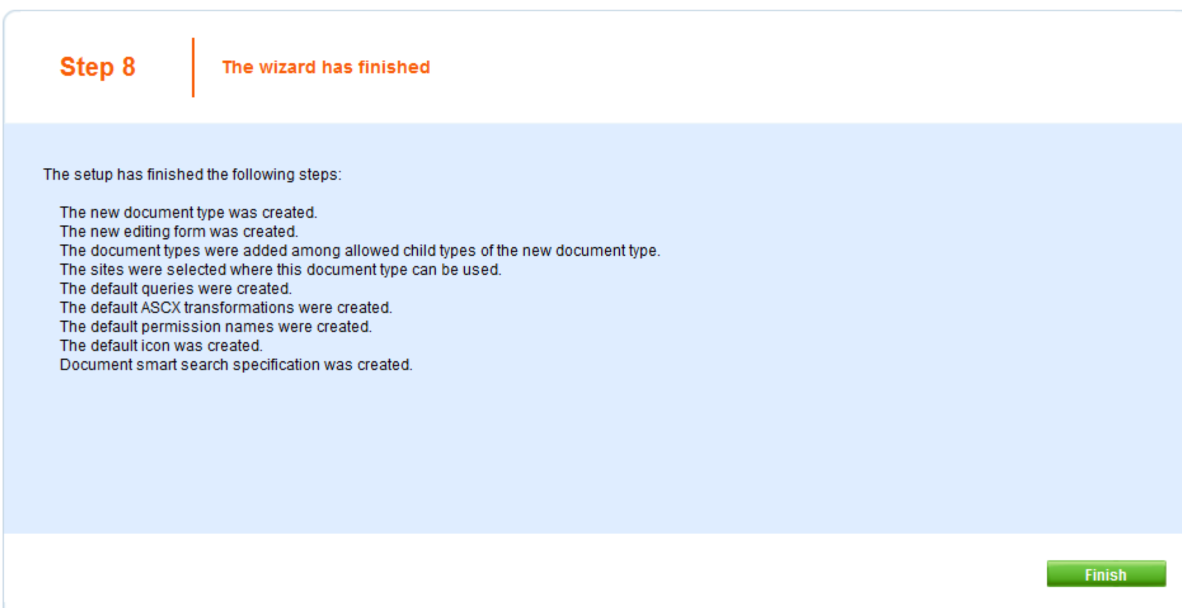
7. In Step 7, you are asked to specify how documents of this type will be indexed and displayed in the search results. For more information on these settings, please refer to this topic. Make your choice and

click **Next**.



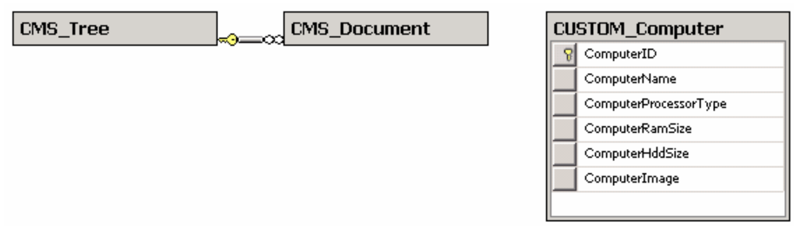
8. The wizard has finished the configuration of the new document type. It has automatically created not only the database table, but also the SQL queries for SELECT, INSERT, UPDATE and DELETE operations and a default transformation.

Click **Finish**. Congratulations, you have learned how to define a new document type.



How the content is stored

As you already know, the new document type *Computer* has its own database table. Each document is stored in three tables: *CMS_TREE* (tree structure), *CMS_Document* (document properties and metadata) and the custom table - in this case *CUSTOM_Computer*:



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

Changing the document type icon

If you create a new document type, documents of this type will appear with the default document icon ( ) in the administration interface. To change the document type icon:

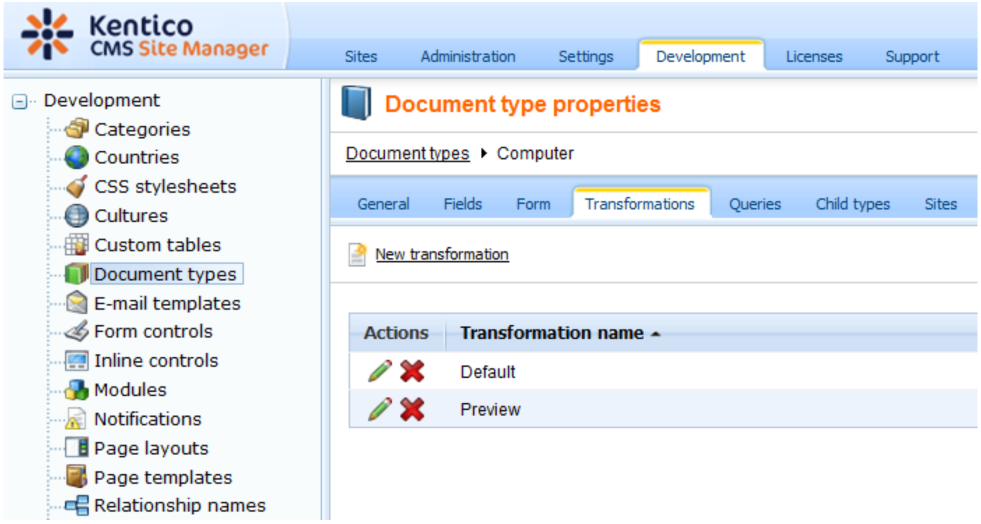
1. go to <*web project folder*>*\App_Themes\Default\Images\DocumentTypeIcons*
2. find your document type's icon named <*namespace*>_<*document type*>*.png*
3. open the file in your image editor and modify it, or create a new image with the same name and format and replace the original one with it

After doing so, all documents of this type should appear with the new icon. You can also change any other document types' icons this way, as all document type icons are stored in the location mentioned above.

7.6.4 Writing transformations

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

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



As you can see, the wizard has created a default transformation. You will use it for the detail view. Click **Edit** ( ) and enter the following code:

```
<h1><%# Eval("ComputerName") %></h1>

<table>
<tr>
<td>
Processor:
</td>
<td>
<%# Eval("ComputerProcessorType") %>
</td>
</tr>

<tr>
<td>
RAM (MB):
</td>
<td>
<%# Eval("ComputerRamSize") %>
</td>
</tr>

<tr>
<td>
HDD (GB):
</td>
<td>
<%# Eval("ComputerHddSize") %>
</td>
</tr>

<tr>
<td>
Image:
```

```
</td>
<td>
<%# GetImage("ComputerImage") %>
</td>
</tr>
</table>
```

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

Now you will create a transformation for the list of computers. Go back to the transformations list and click the **New transformation** link. Enter the following values:

- **Transformation name:** preview
- **Transformation type:** ASCX (it is also possible to use XSLT, but you will not use it now)

Enter the following transformation code:

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

Click **OK**.

Please note how the link to the document is created:

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

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

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

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

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

Please note: Should you want to use the XSLT transformation, it can be used for the **XSLT viewer (CMSViewer)** web part. Otherwise it won't work.

Transformations for multilingual websites

In some cases, you may need to display different texts in transformations, based on the currently selected language. If you are using the built-in multilingual support, you can achieve this by creating another transformation with its name ending with culture code.

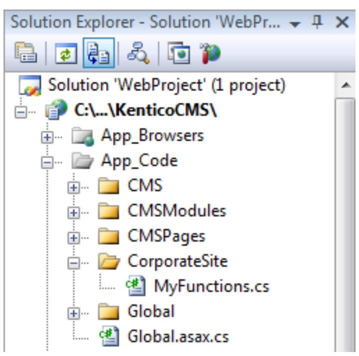
Example:
English (default language) transformation code name: *cms.news.detail*
French transformation code name: *cms.news.detail_fr-fr*

When you switch the content language to French, the French transformation will automatically be used in this case.

7.6.5 Adding custom functions to transformations

In many cases, you may need to process values and display them in a different format or add custom conditions. The following example shows you how to create a custom function that will return the first N characters of a text and how to use it in a transformation.

Open the web project in Visual Studio. Create a new folder under the **App_Code** folder and call it as your site code name. Right-click the folder and choose **Add New Item**. Choose to add a new Class and call it **MyFunctions.cs** (custom transformation functions can be developed only in C# at this moment).



Paste the following code to the *MyFunctions.cs* file:

[C#]

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
```

```
/// <summary>
/// Summary description for MyFunctions
/// </summary>
public static class MyFunctions
{
        public static string TrimText(object txtValue, int leftChars)
        {
            if (txtValue == null | txtValue == DBNull.Value)
            {
                return "";
            }
            else
            {
                string txt = (string) txtValue;
                if (txt.Length <= leftChars)
                {
                    return txt;
                }
                else
                {
                    return txt.Substring(0, leftChars) + "...";
                }
            }
        }
}
```

Please note: The function must be defined as **static** so that you can easily call it from the transformation.

Then, go to **Site Manager -> Development -> Document types -> News -> Transformations**. Edit the **preview** transformation and change its code like this:

[C#]

```
<b><a href="<%# GetDocumentUrl() %>">
<%# Eval("NewsTitle") %></a></b> (<%# GetDate("NewsReleaseDate") %>)<br/>
<i>
<%# MyFunctions.TrimText(Eval("NewsSummary"), 10) %>
</i>
<br/>
```

Click **OK** to save. Go to the live site and see the page with news listing. As you can see, the news summary text is truncated to the first 10 characters.

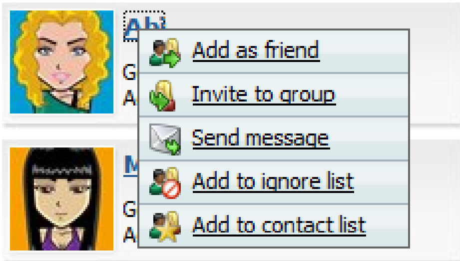
In this topic, you have learned how to write your own transformation methods.

7.6.6 Context menus in transformations

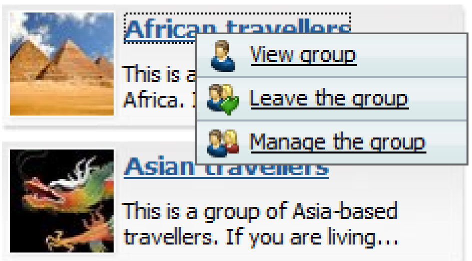
When writing transformations for a web part displaying **Users** or **Groups**, you can enclose the transformation in a menu container control, which ensures displaying context menus after right-clicking on a user or group. You can see a live example of these context menus on the sample **Community**

Starter site:

In the screenshot below, you can see the context menu displayed when you right-click one of the users listed in the **Members** section:



The following screenshot shows the context menu displayed when you right-click one of the groups listed in the **Groups** section:



How is this achieved? As you can see when you view the transformation code used in the **Users viewer** or **Groups viewer** web parts, you need to have enclosed your transformation in the **cms:usermenucontainer** or **cms:groupmenucontainer** control:

Users

```
<cms:usermenucontainer runat="server" ID="userMenuElem" MenuID="userContextMenu"
Parameter='<%# Eval("UserID").ToString() %>' ContextMenuCssClass="UserContextMenu"
 >

... transformation code ...

</cms:usermenucontainer>
```

Groups

```
<cms:groupmenucontainer runat="server" ID="groupMenuElem" MenuID
="groupContextMenu" Parameter='<%# Eval("GroupID").ToString() %>'
ContextMenuCssClass="UserContextMenu" >

... transformation code ...

</cms:groupmenucontainer>
```

Modifying context menu design

The default controls used for context menus are stored in **<web project>\CMSAdminControls\ContextMenus**:

- **GroupContextMenu.ascx**
- **UserContextMenu.ascx**

These two controls are used automatically for the Group or User context menus. If you want to modify the design of the context menus, you can edit these controls in Visual Studio.

You can also develop your custom controls for this purpose. In this case, you need to include the **MenuControlPath** parameter in the *cms:usermenucontainer* or *cms:groupmenucontainer* controls in the transformation and set its value to the path to your control:

```
<cms:groupmenucontainer runat="server" ID="groupMenuElem" MenuID
="groupContextMenu" Parameter='<%# Eval("GroupID").ToString() %>'
ContextMenuCssClass="UserContextMenu" MenuControlPath
="~\CMSAdminControls\ContextMenus\MyGroupContextMenu.ascx" >

... transformation code ...

</cms:groupmenucontainer>
```

7.7 E-mail templates

7.7.1 E-mail templates

Kentico CMS sends automatic system e-mails for various purposes, such as for example workflow notification. The content of these e-mails is determined by templates according to the type of a given e-mail. E-mail templates can be modified at **Site Manager -> Development -> E-mail templates**, which allows you to alter the e-mails sent by the system to match a certain design and/or language.

The following Kentico CMS modules and features send e-mails based on these templates:

- Blogs
- Message boards
- E-commerce (E-commerce Guide -> Configuration settings -> Customizing invoice and e-mail templates)
- Forums
- Friends
- Groups
- Membership - Registration
- Messaging
- Newsletters
- Project management
- Workflow

There are two types of e-mail templates: global and site-specific. If a site-specific template of a certain type is not defined, the global template is used instead.

An e-mail template has the following properties:

Display name	The name of the template displayed in the user interface.
Code name	The name of the template used in code.
From	E-mail address that will be used as the sender (From) address of the e-mail.
Cc	E-mail addresses of copy recipients.
Bcc	E-mail addresses of blind copy recipients. These will get a copy of the e-mail, but won't see the addresses of other recipients in the mail.
Subject	Subject of the e-mail.
HTML version	HTML code of the e-mail template.
Plain text version	Plain text of the e-mail template.

Example of the HTML version of a template:

```
<html>
       <head>
       </head>
       <body style="FONT-SIZE: 12px; FONT-FAMILY: arial">
              <p>
              This an automatic notification sent by Kentico CMS. The following
document is waiting for your approval. Please sign in to Kentico CMS Desk and
approve it.
              </p>
              <p>
              <b>Document:</b> <a href="{%applicationurl%}/cmsdesk/default.aspx?
expandidpath={%idpath%}&expandmode=publish">                {%documentname%}</a>
              <br>
              <b>Last approved by:</b> {%approvedby%}
              <br>
              <b>Last approved when:</b> {%approvedwhen%}
              <br>
              <b>Original step:</b> {%originalstep%}
              <br>
              <b>Current step:</b> {%currentstep%}
              <br>
              <b>Comment:</b>
              <br>
              {%comment%}
              </p>
       </body>
</html>
```

Please note that e-mail template properties may contain macro expressions, such as those marked in the code above, that are resolved (merged) when the e-mail is sent. The use of macros is necessary to ensure that individual e-mails contain information relevant to the situation that caused them to be sent.

7.8 Form controls

7.8.1 Overview

Form controls allow you to use custom field types in editing forms that are based on Kentico FormEngine. These include:

- **Document editing forms (Form tab)**
- **Web part properties**
- **BizForms**
- **Forms using system tables defined in Site manager -> Development -> System tables**
- **Forms using custom tables defined in Site manager -> Development -> Custom tables**
- **Report parameters**

Form controls are standard ASCX controls that inherit from the **CMS.FormControls. FormEngineUserControl** class.

Managing form controls

You can register form controls in the catalog at **Site Manager -> Development -> Form controls**. Form controls have the following properties:

Control display name	The name of the control displayed in the user interface.
Control code name	The name of the control used in code.
Control file name	Relative path to the ASCX control starting from ~/ *CMSFormControls* folder. Example: countryselection.ascx
Use control for text	Indicates if the control can be used for text values.
Use control for long text	Indicates if the control can be used for long text values.
Use control for integer	Indicates if the control can be used for integer values.
Use control for decimal	Indicates if the control can be used for decimal values.
Use control for date-time	Indicates if the control can be used for date-time values.
Use control for boolean	Indicates if the control can be used for boolean (bit) values.
Use control for file	Indicates if the control can be used for files (attachments).
Use control for GUID	Indicates if the control can be used for GUID values.
Use control for visibility	Indicates if the control can be used as a field visibility control. See the Development -> Membership, permissions and security -> Custom fields visibility chapter for more details.
Show control in BizForms	Indicates if the control should be offered among field types in the simple mode of the **Fields** tab of the BizForms module. If you enable this box, you also need to enter the following

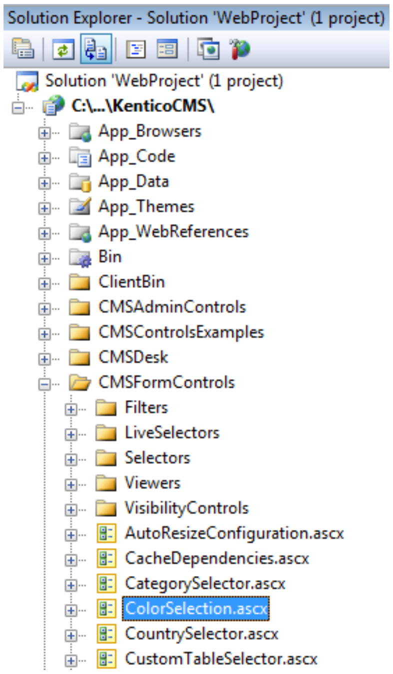
	values: • **Default data type** - the data type of the field that will be used by default when a user chooses to create a new field of this type. • **Column size** - default size of the database column (applies only to the Text data type).
Show control in document types	Indicates if the control should be offered among field types in the document types field editor.
Show control in system tables	Indicates if the control should be offered among field types in the system tables field editor.
Show control in custom tables	Indicates if the control should be offered among field types in the custom tables field editor.
Show control in controls (web parts and inline controls)	Indicates if the control should be offered among field types in the web part property editor.
Show control in reports	Indicates if the control should be offered among field types in the report parameter field editor.

7.8.2 Developing form controls

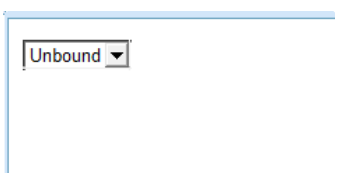
The following example shows how to create a form control that will allow users to choose a color from a drop-down list.

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 **CMSFormControls** folder and choose **Add New Item**. Choose to create a new **Web User Control** and call it **ColorSelection.ascx**.



3. Edit the ColorSelection.ascx user control on the **Design** tab. Drag and drop a **DropDownList** control onto the form:



4. Edit the properties of the DropDownList and change its **ID** to *drpColor*.

5. Switch to the code behind.

6. Change the following line

[C#]

```
public partial class CMSFormControls_ColorSelection : System.Web.UI.UserControl

to

public partial class CMSFormControls_ColorSelection : CMS.FormControls.
FormEngineUserControl
```

[VB.NET]

```
Partial class CMSFormControls_ColorSelection Inherits SystemWeb.UI.UserControl
```

```
to

Partial class CMSFormControls_ColorSelection Inherits CMS.FormControls.
FormEngineUserControl
```

This ensures that our user control inherits from the **FormEngineUserControl** class and can use its standardized properties.

7. Modify the code behind like this:

[C#]

```
protected void Page_Load(object sender, EventArgs e)
{
     // Ensure drop down list options
     EnsureItems();
}

/// <summary>
/// Gets or sets field value, color hexa code in this case.
/// </summary>
public override Object Value
{
     get
     {
         return drpColor.SelectedValue;
     }
     set
     {
         // Ensure drop down list options
         EnsureItems();
         drpColor.SelectedValue = System.Convert.ToString(value);
     }
}

/// <summary>
/// Returns an array of values of any other fields returned by the control.
/// </summary>
/// <returns>It returns an array where first dimension is attribute name and the
second dimension is its value.</returns>
public override object[,] GetOtherValues()
{
object[,] array = new object[1, 2];
array[0, 0] = "ProductColor";
array[0, 1] = drpColor.SelectedItem.Text;
return array;
}

/// <summary>
/// Returns true if some color is selected. If not, it returns false and displays
an error message.
/// </summary>
public override bool IsValid()
{
    if ((string)Value != "")
```

```
        {
            return true;
        }
        else
        {
            // Set form control validation error message
            this.ValidationError = "Please choose some color.";
            return false;
        }
    }

    /// <summary>
    /// Ensures that the DropDownList contains color options.
    /// </summary>
    protected void EnsureItems()
    {
        if (drpColor.Items.Count == 0)
        {
            drpColor.Items.Add(new ListItem("(select color)", ""));
            drpColor.Items.Add(new ListItem("red", "#FF0000"));
            drpColor.Items.Add(new ListItem("green", "#00FF00"));
            drpColor.Items.Add(new ListItem("blue", "#0000FF"));
        }
    }
```

[VB.NET]

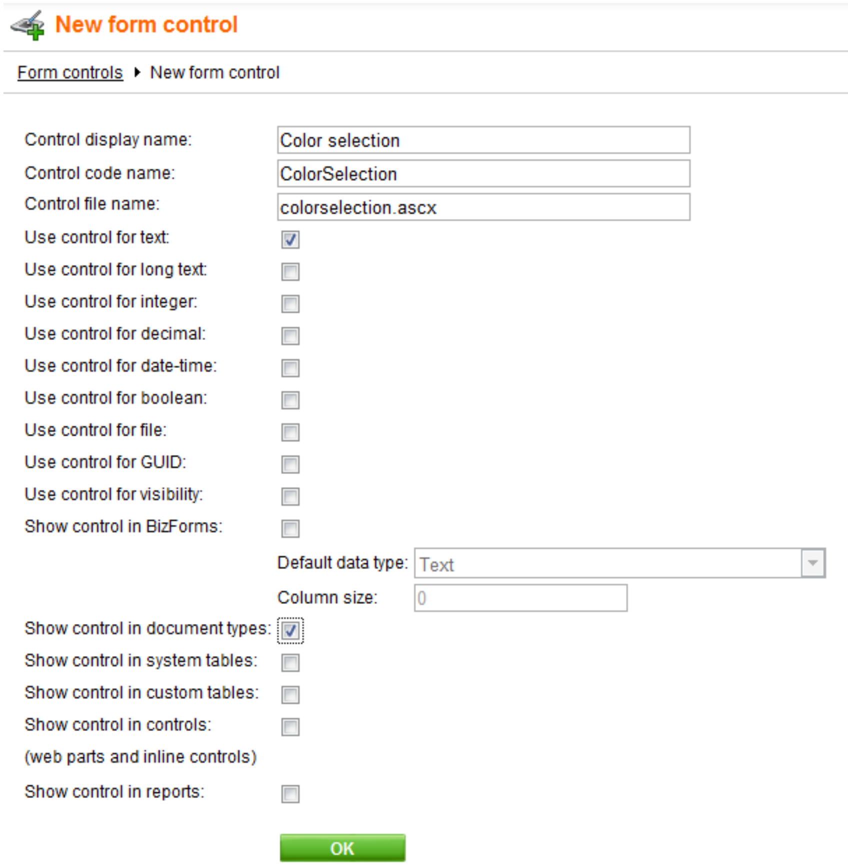
```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    ' Ensure drop down list options
    EnsureItems()
End Sub

''' <summary>
''' Gets or sets field value, color hexa code in this case.
''' </summary>
Public Overrides Property Value() As Object
    Get
        Return drpColor.SelectedValue
    End Get
    Set(ByVal value As Object)
        EnsureItems()
        drpColor.SelectedValue = System.Convert.ToString(value)
    End Set
End Property

''' <summary>
''' Returns an array of values of any other fields returned by the control.
''' </summary>
''' <returns>It returns an array where first dimension is attribute name and the
second dimension is its value.</returns>
Public Overrides Function GetOtherValues() As Object(,)
    Dim arr(0, 1) As Object
    arr(0, 0) = "ProductColor"
    arr(0, 1) = drpColor.SelectedItem.Text
    Return arr
```

```
End Function

''' <summary>
''' Returns true if some color is selected. If not, it returns false and displays
an error message.
''' </summary>
Public Overrides Function IsValid() As Boolean
    If CType(Value, String) <> "" Then
        Return True
    Else
        ' Set form control validation error message
        Me.ValidationError = "Please choose some color."
        Return False
    End If
End Function

''' <summary>
''' Ensures that the DropDownList contains color options.
''' </summary>
Public Sub EnsureItems()
    If drpColor.Items.Count = 0 Then
        drpColor.Items.Add(New ListItem("(select color)", ""))
        drpColor.Items.Add(New ListItem("red", "#FF0000"))
        drpColor.Items.Add(New ListItem("green", "#00FF00"))
        drpColor.Items.Add(New ListItem("blue", "#0000FF"))
    End If
End Sub
```

8. Go to **Site Manager -> Development -> Form controls** and click **New form control**. Enter the following values:

- **Control display name**: Color selection
- **Control code name**: colorselection
- **Control file name**: colorselection.ascx

Check the **Use control for text** box, check the **Show control in document types** box and click **OK**.



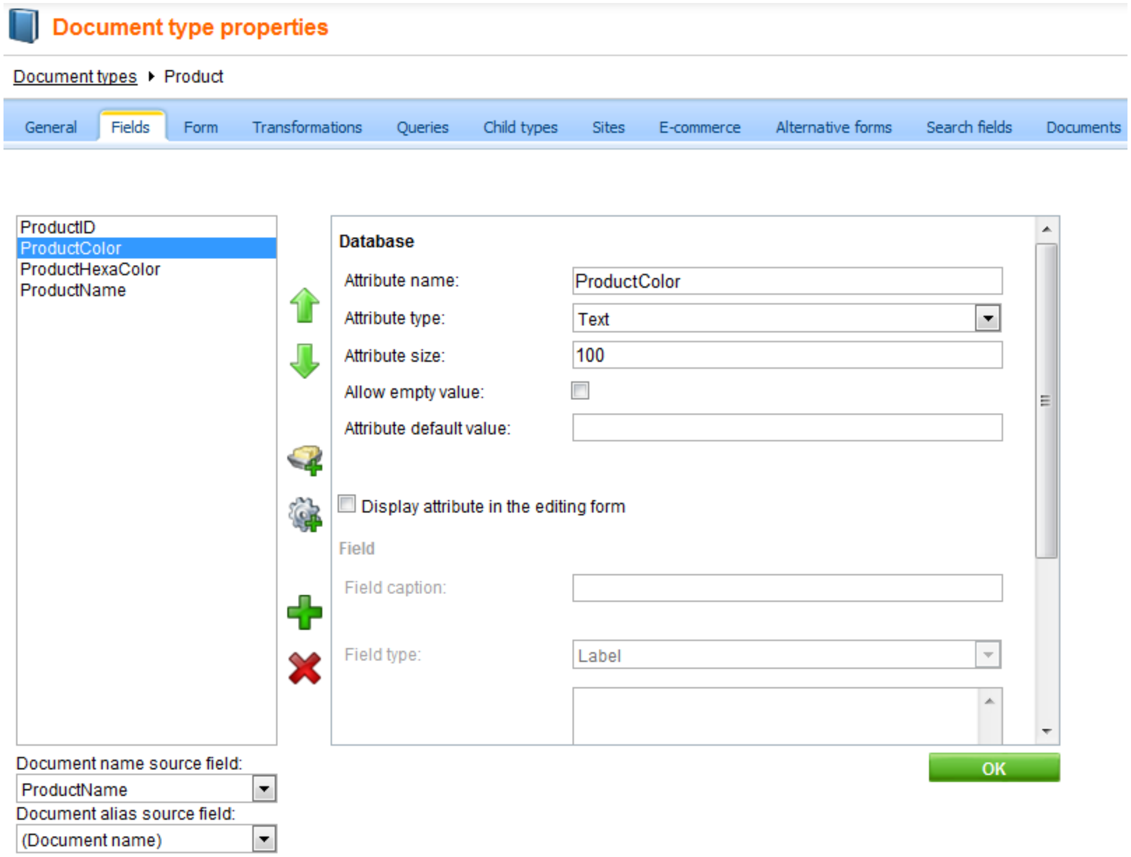
9. Now we will test this control in some document editing form. Go to **Site Manager -> Development -> Document types** and edit ( ) the **Product** document type. Click the **Fields** tab and add 2 new fields with the following properties:

- **Attribute name**: ProductColor
- **Attribute type**: Text
- **Attribute size**: 100
- **Display attribute in editing form**: false (the field will be set via the **GetOtherValues()** method of the ColorSelection control)

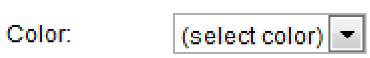
Click **OK** to save the new field

- **Attribute name**: ProductHexaColor
- **Attribute type**: Text
- **Attribute size**: 100
- **Field caption**: Color
- **Field type**: Color selection

Click **OK** to save the new field



10. Go to **CMS Desk -> Content** and create a new product in the **Products** section. The new form control will be displayed like this:



If you do not choose any color, the validation error label is displayed.

Getting and setting values of other fields using the API

You can get and set the values of the other fields by getting their reference like this:

[C#]

```
TextBox txtBox = (TextBox) Form.FieldControls["productprice"];
```

[VB.NET]

```
Dim txtBox as TextBox = CType(Me.Form.FieldControls("productprice"),
TextBox)
```

Please note that the field name must be written **in lowercase**.

When you get the reference to the control, you can get or set its value (for example: txtBox.Text = "150")

If you want to change the value of some other control **before it's loaded**, you need to place the code inside the **PageLoad** method of your form control.

If you want to change the value of some other control **before it's saved**, you need to place the code inside the **IsValid** method of your form control.

You can also use the **this.DataDR (me.DataDR)** property of the form control that provides the DataRow object with data of the current form.

7.9 Inline controls

7.9.1 Overview

Inline controls are user controls (ASCX) that can be placed into the text of editable regions using a special expression in format *%%control:MyUserControl%%,* where *MyUserControl* is the name of the inline control. The system dynamically loads the controls when the page is displayed on the live site.

The controls may contain any functionality, such as "polls", "latest news", "mortgage calculator", "travel destination search", etc. The advantage of inline controls is that any content editor can place them anywhere into the text without programming knowledge.

How to insert inline controls into text

Inline controls can be inserted into text using the **Insert inline control** () button on the WYSIWYG editor toolbar. If the text is not edited in the WYSIWYG editor, you can insert inline controls by typing the *%%control:MyUserControl%%* expressions manually. Only controls that have been registered in the system and assigned to the currently edited site can be inserted. This can be done at **Site Manager -> Development -> Inline controls**.

The inline control may also have a single parameter. In this case, you must use one of the following expression formats:

- %%control:BizFormControl?form1%%
- {^BizFormControl|form1^}
- {^BizFormControl|(formname)form1^}

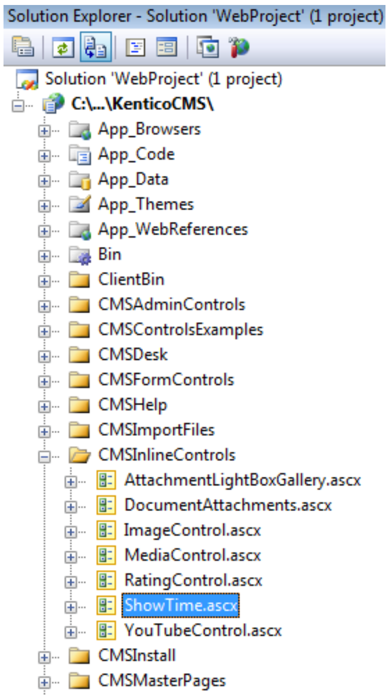
To learn how to create a new inline control and add it to the system, please see the How to develop inline controls topic.

7.9.2 How to develop inline controls

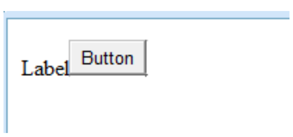
This topic will show you an example of inline control development. We will create a simple control that will display the current time when a button is clicked.

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 **CMSInlineControls** folder and choose **Add New Item**. Choose to create a new **Web User Control** and call it **ShowTime.ascx**.



3. Switch to the **Design** tab and drag and drop a **Label** control and a **Button** control onto the form:



4. Double-click the Button control and enter the following code into the **Button1_Click** method. It will ensure the displaying of the current date and time by the label control.

[C#]

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

[VB.NET]

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

5. Change the following line:

[C#]

```
public partial class CMSInlineControls_ShowTime : System.Web.UI.UserControl

to

public partial class CMSInlineControls_ShowTime : CMS.ExtendedControls.
InlineUserControl
```

[VB.NET]

```
Partial Class CMSInlineControls_ShowTime
    Inherits System.Web.UI.UserControl

to

Partial Class CMSInlineControls_ShowTime
    Inherits CMS.ExtendedControls.InlineUserControl
```

What you did

You have changed the user control so that it inherits from the **InlineUserControl** class. It allows you to access the parameter of the control in the next step.

6. Add the following code to the **Page_Load** method of the control:

[C#]

```
Button1.Text = this.Parameter;
```

[VB.NET]

(The **Page_Load** method is not generated by default in VB.NET)

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    Button1.Text = Parameter
End Sub
```

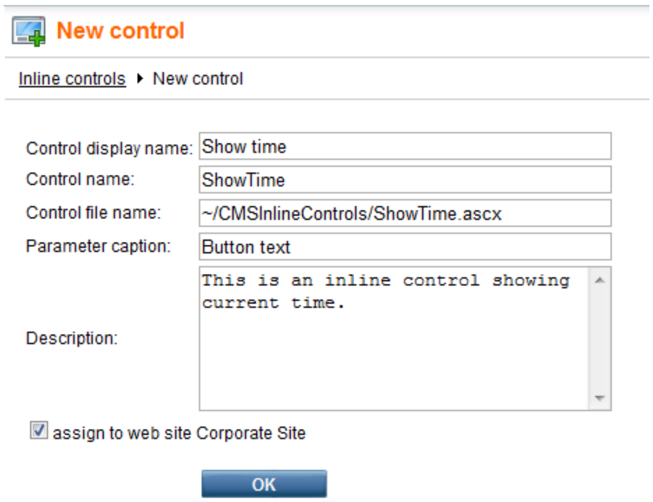
What you did

This code sets the Button1 caption based on the parameter of the inline control.

7. Save the changes.

8. Go to **CMS Site Manager -> Development -> Inline controls**. Click **New control** and enter in the following values:

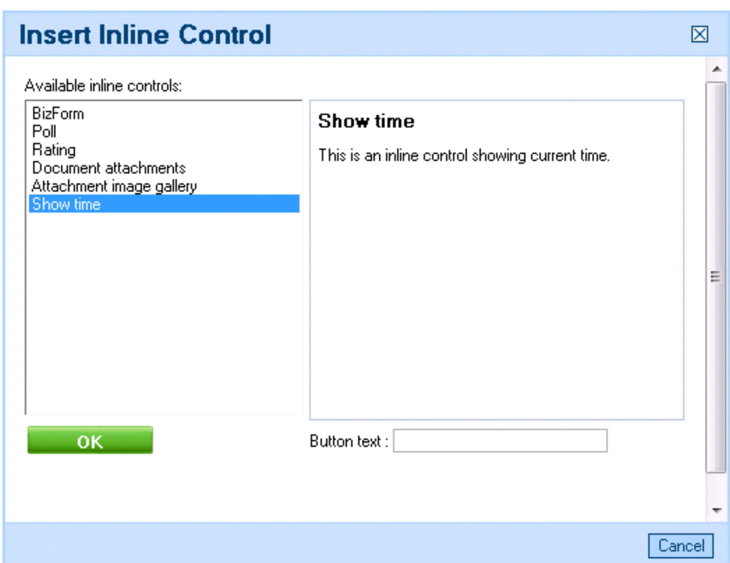
- **Control display name:** Show time
- **Control name:** ShowTime (the name of the user control without the extension)
- **Control file name:** ~/CMSInlineControls/ShowTime.ascx
- **Parameter caption:** Button text



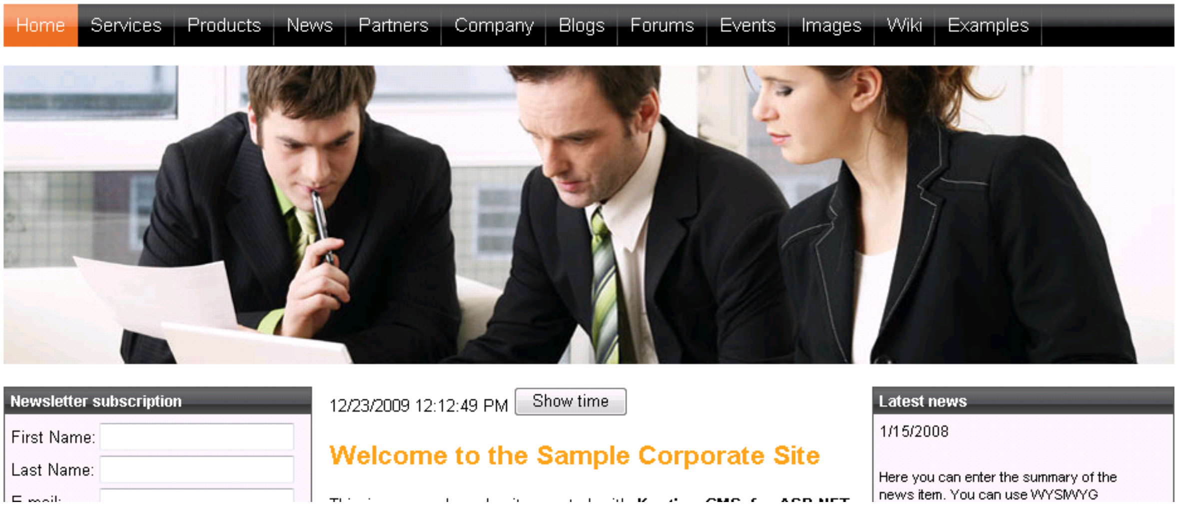
Click **OK**.

9. Now click the **Sites** tab and check the boxes for all sites where content editors should be able to insert this control. Click **OK**.

10. Go to CMS Desk, edit some page with editable regions and click the **Insert Inline Control** ( ) button. Select the control and set the **Button text** value to *Show time*. Click **OK**. The special expression is now inserted into the text.



11. Click **Save** to save changes and click **Live site** to see the live version of the page. The user control is now displayed inside your text. The button has the caption you have specified. When you click the button, the label displays the current date and time:



7.10 Membership, permissions and security

7.10.1 Security model overview

Kentico CMS provides a flexible security model that allows you to configure granular access permissions for content and modules.

The security model consists of:

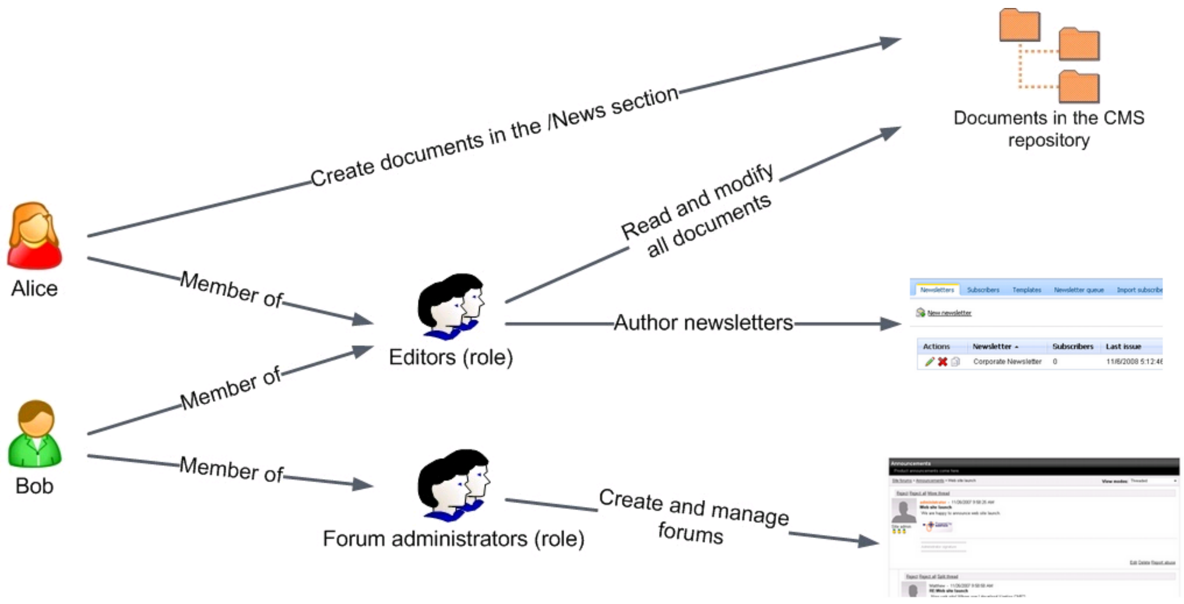
- users (shared among websites)
- roles (specific for websites)
- module permissions
- document permissions
- UI personalization

User, role and global permissions can be managed at two levels:

- In **Site Manager -> Administration**, where global administrators can edit all data.
- In **CMS Desk -> Administration**, where local administrators can edit only data related to the current website (the current website is recognized by the current domain).

Relationships between users, roles and permissions

The following figure shows how users are assigned to roles and how permissions for documents and modules are granted to users and roles:



Users can be members of any number of roles. Permissions for particular documents in the CMS repository can be granted to them. If you want to grant permissions for some module to a user, you need to make the user a member of some role and grant the permissions to the role (i.e. permissions for modules cannot be granted to users directly).

Roles in Kentico CMS are fully customizable. It means you're not limited to some predefined set of roles. Instead, you can define your own roles with custom sets of permissions.

If a user is a member of multiple roles, their **permissions for modules** are calculated as a sum of all permissions granted to all roles.

If **permissions for documents** in the CMS repository are granted to both a user and their roles, document permissions are calculated as a sum of all permissions granted to the user and to all roles. If a user or some of their roles are **denied to make some action** (such as modify document), then the result is always "denied" for the given permission, even if some of the roles are allowed to perform the action.

7.10.2 User management

A user can be a member of any number of roles and can be assigned to any number of websites.

There are two important attributes of the user account:

- **Is editor** – the user can access the **CMS Desk** interface; this attribute doesn't implicate any particular permissions - it only differentiates between the site editors and "registered users" who only access the live website and its secured areas and thus provides an extra security layer; the user can access CMS Desk for all sites to which they are assigned on the **Sites** tab
- **Is global administrator** – the user is authorized to perform all operations and their access cannot be denied by permissions or otherwise limited; global administrators are the only users who can use the **Site Manager** interface

Global administrators

Global administrators are the only users who can manage site settings and all development tools. Their permissions cannot be denied or limited – they have access to all features and data.

Local administrators cannot modify global administrator accounts.

Default user accounts

The following default user accounts are available:

- **Administrator** – user with full permissions.
- **Public** – user that represents an anonymous visitor of the site.

Creating a new user

New user accounts are typically created when a user goes through registration on the live site. However, you can also create accounts manually in **Site manager -> Administration -> Users** or **CMS Desk -> Administration -> Users**. Click the **New user** link and enter the following properties into the displayed form:

- **User name** - the user's user name (login); it must be unique across all websites in the system
- **Full name** - user's full name (first name, middle name and last name)
- **E-mail** - user's e-mail address
- **Enabled** - indicates if the user account is enabled and the user can sign in
- **Is editor** - indicates if the user is authorized to sign in to CMS Desk; it's used to differentiate users who are only allowed to visit member areas of the website from content editors who can use the CMS Desk user interface; this provides an extra security layer
- **Password** - user's password
- **Confirm password** - user's password again for confirmation

Editing user properties

You can edit user properties in **Site manager -> Administration -> Users** -> click the **Edit** () icon of the chosen the user.

General properties

The following properties can be set on the **General** tab:

- **User name** - the user's user name (login); it must be unique across all websites in the system
- **Full name** - user's full name (first name, middle name and last name)
- **First name** - user's first name
- **Middle name** - user's middle name
- **Last name** - user's last name
- **E-mail** - user's e-mail address

- **Enabled** - indicates if the user account is enabled and the user can sign in

- **Is editor** - indicates if the user is authorized to sign in to CMS Desk; it's used to differentiate users who are only allowed to visit member areas of the website from content editors who can use the CMS Desk user interface; this provides an extra security layer
- **Is global administrator** - indicates if the user is a global administrator; global administrators have full permissions for all features and data across the system and are not affected by permission settings for particular modules
- **Is external user** - this attribute is used when you are using an integration with an external user database
- **Is domain user** - indicates if the user was imported from Active Directory
- **Is hidden** - if true, the user is not visible on the site (e.g. on-line user monitoring, repeaters displaying users, etc.)
- **Disable site manager -** this option is available only when editing a global administrator, but not when a global administrator is editing their own account; if enabled, the user is still a global administrator, but cannot access Site Manager, i.e. can only perform actions in CMS Desk

- **Preferred content culture -** preferred culture in which the content is displayed to the user
- **Preferred user interface culture -** preferred culture in which the users wants to see the user interface (CMS Desk and Site Manager)

- **Created -** date and time when the user account was created
- **Last logon -** date and time when the user last logged in
- **Last logon information** - information about the IP address and browser agent of the user's last logon

- **Starting alias path** - the starting alias path of the content tree in CMS Desk -> Content; if you specify this value, the user is not allowed to browse other sections of the website in the content tree; please note that this feature is only intended for better usability and it doesn't ensure security control - if you need to establish access rights for a given user, grant appropriate document permissions (Properties -> Security) to them

Impersonation

Global administrators can also see the **Log in as this user** link at the top of this tab. By clicking this link, the administrator gets logged in as the currently edited user and gets redirected depending on the type of user:

- **Global administrator** - if you log in as some other global administrator, you will stay on the **General** tab
- **Editor** - if you log in as some editor (a user with the **Is editor** option enabled), you will be redirected to CMS Desk
- **Standard user** - if you log in as a standard user, you will be redirected to the title page of the live site

In **Site Manager -> Administration -> Event log**, any changes made while impersonated will be logged with under the user name in format <user name> (<original user name>) where the original user is the administrator who clicked the **Log in as this user** link.

Password

Here you can change the user's password:

- **Password** - user's password

- **Confirm password** - user's password again for confirmation

Please be aware that this tab is hidden if the edited user is authenticated using either an external user database or Active Directory, i.e., if the user has the **Is external user** or **Is domain user** property enabled on the **General** tab of the user editing interface and the application is configured to use Windows authentication.

Settings

On the **Settings** tab, you can edit the following properties of the user:

- **User nick name** - nick name of the user used in website forums, on the user's profile, etc.
- **User picture** - user's avatar image; this image will be used in forums and on user's profile; you can either upload an image or select a pre-defined avatar
- **User signature** - user's signature that will be used below the user's forum posts
- **Description** - optional text describing the user
- **URL referrer** - URL from that the user came to the site when they performed registration
- **Campaign** - the campaign from which the user came to your website before they registered; see the Modules -> Web analytics -> Tracking marketing campaigns topic for details
- **Messaging notification e-mail** - notifications about new messages received in the messaging module will be sent to this e-mail address

- **Time zone** - user's time zone; if set, this time zone will be used where applicable instead of the site time zone
- **Badge** - user's badge; depends on the number of gained activity points

- **User activity points** - number of user's activity points; these points are gained for forum posts, message board posts, blog posts and blog post comments
- **Live ID** - user's Live ID token; this is a hexadecimal number that the user is identified by when logging-in via Windows Live ID
- **Facebook user ID** - user's Facebook user ID; it is used when the user is logging in via Facebook Connect
- **OpenID** - user's OpenID; it is used when the user is logging in via OpenID

- **Activation date** - date of the user's account activation
- **Activated by user** - user who activated this user's account
- **Registration info** - user's IP and browser agent detected on registration

- **Gender** - user's gender
- **Date of birth** - user's date of birth

- **Skype account** - user's Skype account
- **Instant messenger** - user's instant messenger; format of values of the field is not strictly required, you may use any string of characters according to your specific needs (e.g. *ICQ: 123456789*)
- **Phone number** - user's phone number; the number may be entered in any format, no validation is applied

- **Waiting for approval** - if checked, the user is waiting for administrator's approval
- **Show splash screen** - determines if splash screen should be displayed to the user when accessing Kentico CMS administration interface

- **Forum posts** - number of user's forum posts
- **Forum comments** - number of user's forum comments

- **Blog comments** - number of user's blog comments
- **Message board posts** - number of user's message board posts

Custom Fields

Here you can edit the custom fields added to the user profile. The custom fields can be defined in **Site Manager -> Development -> System tables -> User**.

Sites

Here you can specify the sites into which the user can sign in with their user name and password. Simply check the appropriate boxes and click **OK** to save the changes. This dialog applies only to logging in to CMS Desk and is intended for separation of content editors responsible for different websites.

Roles

Here you can add or remove user from roles. First, you need to select the site (since roles are always connected with a single website) and then check the appropriate boxes and click **OK**.

Please note: if you cannot see any sites in the listbox, you may need to assign the user to some site on the **Sites** tab.

Departments

Here you can specify the E-commerce module departments the user is authorized to manage products in.

Notifications

On this tab, you can see a list of all notification subscriptions of the currently edited user. You can **Delete** (✖) subscriptions in the list, which unsubscribes the user from receiving notifications.

Categories

This tab displays a list of the user's custom categories. Each of the categories can be edited (✏) or deleted (✖).

By clicking the **New category** link, you can create a new category that will behave the same way as if it was created by the user in **CMS Desk -> Edit -> Properties -> Categories**.

The following details will be required when creating a new category:

- **Display name** - name of the category displayed in the user interface
- **Code name** - name of the category used in website code

Friends

On this page, you can manage the currently edited user's friends.

Subscriptions

On this tab, you can manage the user's subscriptions to newsletters and notifications about new blog posts and message board messages.

Languages

On this tab, you can specify which cultural versions of documents can be edited by the user. You have the following options:

- **User can edit all languages -** if selected, the currently edited user can edit documents in all language versions of all sites in the system
- **User can edit following languages -** if selected, you can specify which language versions can be edited by the user by checking the check-boxes in the list of language versions; this can be set separately for each site in the system using the **Select site** drop-down list

7.10.3 Role management

Roles are assigned to a particular (single) website. A role can have any number of members.

You can manage roles in **Site Manager -> Administration -> Roles**.

Creating a new role

Go to **Site Manager -> Administration -> Roles** and click **New role** .Enter the following values:

- **Role display name** - The name of the role displayed to the users.
- **Role code name** - The name of the role used in the code.
- **Role description** - Optional description.
- **Is domain role** - Indicates if the role was imported from Active Directory.

Editing a role

Go to **Site Manager -> Administration -> Roles** and click the **Edit** () button of the appropriate role.

General

- **Role display name** - The name of the role displayed to the users.
- **Role code name** - The name of the role used in the code.
- **Role description** - Optional description.
- **Is domain role** - Indicates if the role was imported from Active Directory.

Users

Here you can add or remove users to/from the current role. If you want to add a user, click the **Add user to role** button. The **Select users** dialog appears. Enter the name of the user or its part and click **Search users**. Now select the listed users in the box (you can use CTRL+left mouse button) and click **Add selected users to role**.

Deleting a role

Go to **Site Manager -> Administration -> Roles** and click the **Delete** (✖) button of the appropriate role.

7.10.4 Username customization

If you want to customize the way usernames are displayed in the administration interface, you can do it by modifying the **GetFormattedUsername** method in **~/AppCode/CMS/Functions.cs**. The method has four overrides and is used to retrieve usernames in the whole administration interface.

Example: Usernames are displayed in the *<full name> (<user name>)* (e.g. *Abigail Woodwarth (Abi)*) format in some parts of the system, e.g. in document **Properties -> General -> Owner**. The following code example shows how you can modify the method to get usernames in format *<user name> [<full name>]* (e.g. *Abi [Abigail Woodwarth]*):

[C#]

```
public static string GetFormattedUserName(string username, string fullname, bool
isLiveSite)
{
    if (!String.IsNullOrEmpty(DataHelper.GetNotEmpty(fullname, "").Trim()))
    {
        return String.Format("{1} [{0}]", fullname, username);
    }
    else
    {
        return username;
    }
}
```

7.10.5 Permissions

7.10.5.1 Permissions overview

Permissions are a way how you can control access to particular sections of Kentico CMS administration interface (modules), documents in the content tree and custom tables.

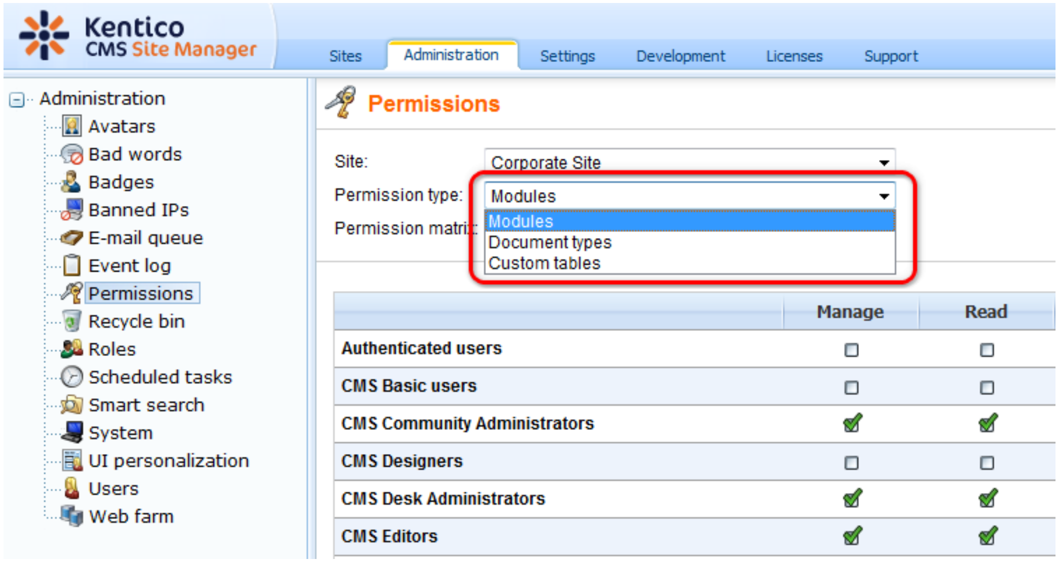
Permissions for roles

Permissions for roles can be configured in **Administration -> Permissions**. Every website has its own set of roles (there are no shared or global roles). These permissions are also assigned to roles, which means every website can use a different configuration of role permissions.

Based on the selection made by the **Permission type** drop-down list, you can choose from the following three types of permissions. Then you need to select the appropriate Permission matrix, which will be offered based on the selected Permission type.

- **Modules** - permissions for specified actions in Kentico CMS modules; you can find details on particular permissions in documentation of respective modules
- **Document types** - permissions applied to all documents of a particular type; these permissions represent one level of the three-level document permissions hierarchy, as described in the Document permissions topic

- **Custom tables** - permissions for the custom tables module, see Modules -> Custom tables -> Security for more info

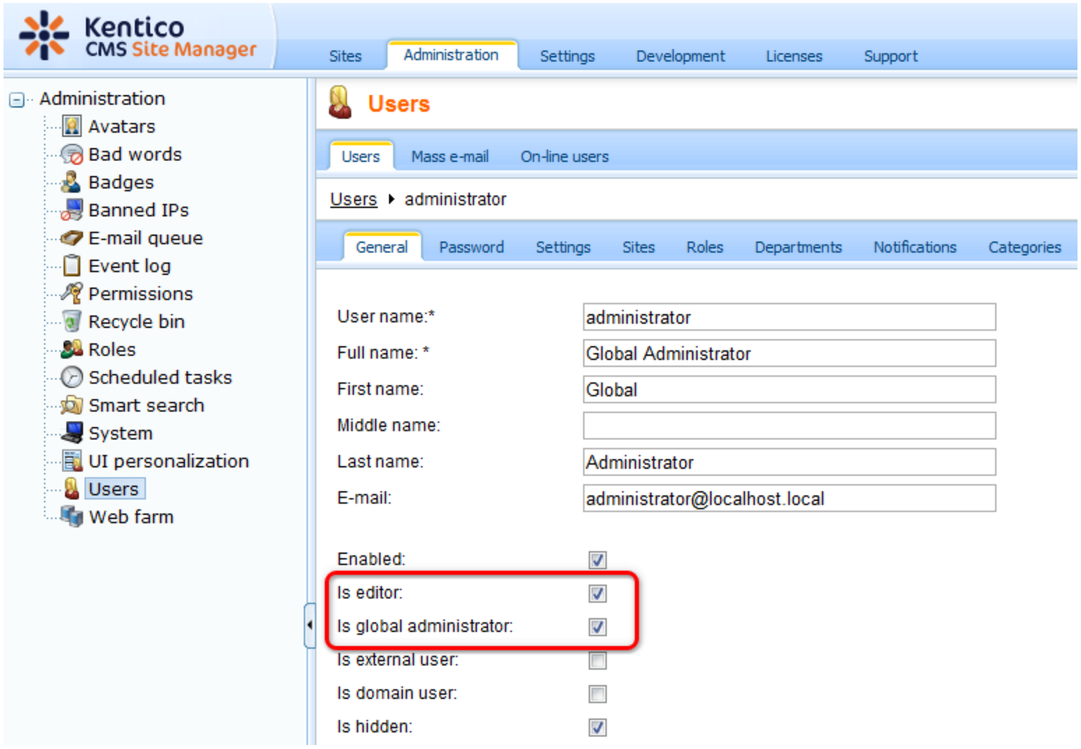


Permission-related settings for users

When editing ( ) a user in **Administration -> Users**, you can enable the following two options for the user. These options have impact on permissions checking and provide an extra security layer:

- **Is global administrator** - the user is authorized to perform all operations and their access can not be denied by permissions or otherwise limited; global administrators are the only users who can use the Site Manager interface

- **Is editor** – the user can access the CMS Desk interface; this attribute doesn't implicate any particular permissions - it only differentiates the site editors and "registered users" who only access the live website and its secured areas and thus provides and extra security layer; the user can access CMS Desk for all sites to which they are assigned on the Sites tab



7.10.5.2 Document permissions

Permissions for access to documents can be configured at three levels:

1. **Permissions for all content -** granted to roles
2. **Permissions for document types -** granted to roles
3. **Document-level permissions -** granted to roles or individual users

Permissions from these three levels are merged together when checking if a user is a permitted to perform an action with a document. For example, to read a document, a user must have the *Read* permission on at least one of the three levels: either on document-level, or for the document's document type, or for all content.

Permissions for all content

In **Site Manger -> Administration -> Permissions**, there is a special permission matrix for controlling access to all documents within the content tree. It is the **Modules -> Content** permission matrix.

The following global permissions can be granted to particular roles:

- **Read** - allows members of the role to view any document in the content tree
- **Modify** - allows members of the role to modify any document in the content tree
- **Check in any document** - authorizes user to perform the Check in or Undo check-out actions on the Properties -> Versions tab of a document
- **Create** - allows members of the role to create any document in the content tree
- **Delete** - allows members of the role to delete any document in the content tree
- **Manage workflow** - allows members of the role to approve/reject any document at any workflow step

- **Destroy** - allows members of the role to destroy any document (delete without undo option)
- **Modify permissions** - manage local permissions of any document
- **Browse tree** - allows members of the role to browse the content tree; if not assigned, the Content tab may not be displayed (unless the role has the Read permission for the *CMS.Root* document type or for the *Root* document (on document level))
- **Design website** - allows members of the role to access the Design tab; please note: although users can make changes only to the current website, the changes may affect other websites if they modify a page template shared among multiple websites

Permissions for document types

Document type permissions allow control of access to all documents of a particular document type in the content tree. These permissions are assigned to roles in **Administration -> Permissions**, by selecting **Permission type:** *Document types* and choosing the document type from the **Permission matrix** drop-down list. All documents of a type will have access limited by the permissions configured for the document type.

You can grant the following document type permissions to particular roles:

- **Read** - read all documents of this type
- **Create** - create documents of this type
- **Modify** - modify all documents of this type
- **Delete** - delete all documents of this type
- **Destroy** - destroy all documents of this type
- **Browse tree** - display child documents of all documents of this type
- **Modify permissions** - manage local permissions of all documents of this type

Document-level permissions

You can also configure permissions on document level, directly in the content tree. These permissions are merged with global permissions for all content (the *Content* module) and global permissions for document types. Document-level permissions are described in detail on the following page.

7.10.5.3 Document-level permissions

You can manage local permissions (i.e. permissions for a particular document or site section) in **CMS Desk -> Content ->** select a document in the content tree **-> Properties -> Security**.

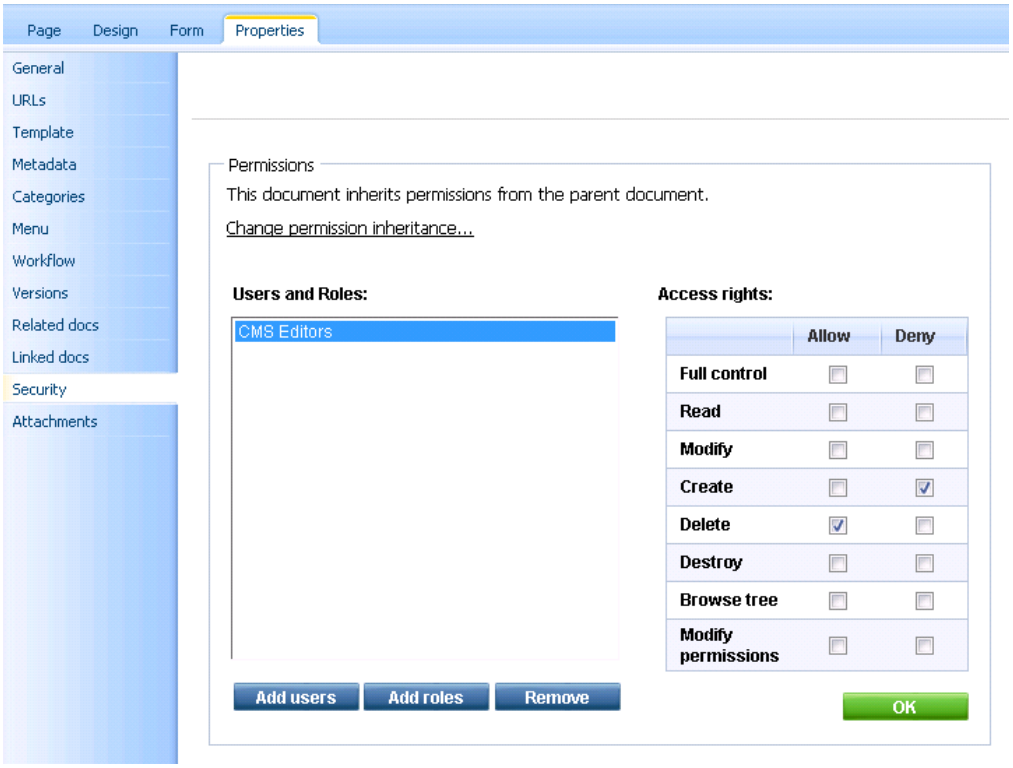
Select the appropriate user or role in the left box. If the user or role is not available in the box, you may need to add them using the **Add users** or **Add roles** button. Now you can choose if the permissions should be "allowed" or "denied".

If you deny a permission, the action will not be allowed even if the user or role has the permission assigned on a global level, i.e. the Deny option overrides global settings for this permission.

You can configure the following permissions:

- **Full control** - perform all operations with the document
- **Read** - read document content
- **Modify** - modify document content, check in, check out
- **Create** - create new documents under this document
- **Delete** - delete this document

- **Destroy** - destroy this document (without undo option)
- **Browse tree** - unfold the current document and see its child documents
- **Modify permissions** - change document permissions



Permission inheritance

You will typically need to set up permissions for site sections, rather than for particular documents. In this case, you grant permission for the main section document to users and these permissions are inherited by all child documents.

Example
Consider the site structure like this:

- Root
  - Home
  - News
  - Products
    - Category 1
    - Category 2

You may want to grant the following permissions to the users:

JohnS	**Marketing manager** John can manage all content.	Grant the **Full control** permission on the root to the user or grant permissions for the **CMS Content** module to some of this user's roles.
MarkJ	**Product manager** Mark can manage only the	Grant the **Browse tree** permission on the root to the user so that they can browse the Products section.

	documents in the /Products section.	Grant the **Read**, **Modify**, **Create**, **Delete**, **Destroy** and **Browse tree** permissions on the /Products document to the user. These permissions are inherited by all child documents under the /Products section. Please note that if you click the /Products/Category 1 document, the Browse tree permission is grayed and disabled. It means that this **permission is inherited** and cannot be removed - you can only **deny** the permission (unless you break inheritance - see below).
AliceM	**Copy writer** Alice can modify the copy of all documents, but Mark prefers to manage the copy of the /Products section by himself only.	Grant the **Read**, **Modify**, **Create**, **Delete** and **Browse tree** permissions for the root to the user. Go to the /Products document and **deny** the **Modify**, **Create**, **Delete** permissions to the user so that Alice cannot modify the copy in the /Products section.

Please note: It's recommended that you configure local permissions for roles and then only assign users to the appropriate roles. In this example, you would first create roles "Marketing manager", "Product manager" and "Copy writer" and then configure their permissions.

Breaking the inheritance

In case you need to break the permission inheritance and configure different permissions for some site section, you need to click the **Change permission inheritance...** link in the **Security** dialog and choose one of the following options:

- **Break inheritance and copy parent permissions** - breaks permission inheritance and adds parent permissions to the document, while original permissions configured for the document are preserved.
- **Break inheritance and remove parent permissions** - breaks inheritance and removes all permissions inherited from the parent, while additional permissions configured for the document will be preserved.

Restoring the inheritance

If you decide to inherit the permissions from the parent again, click the **Change permission inheritance...** link in the **Security** dialog and then click **Restore inheritance to parent document permissions**.

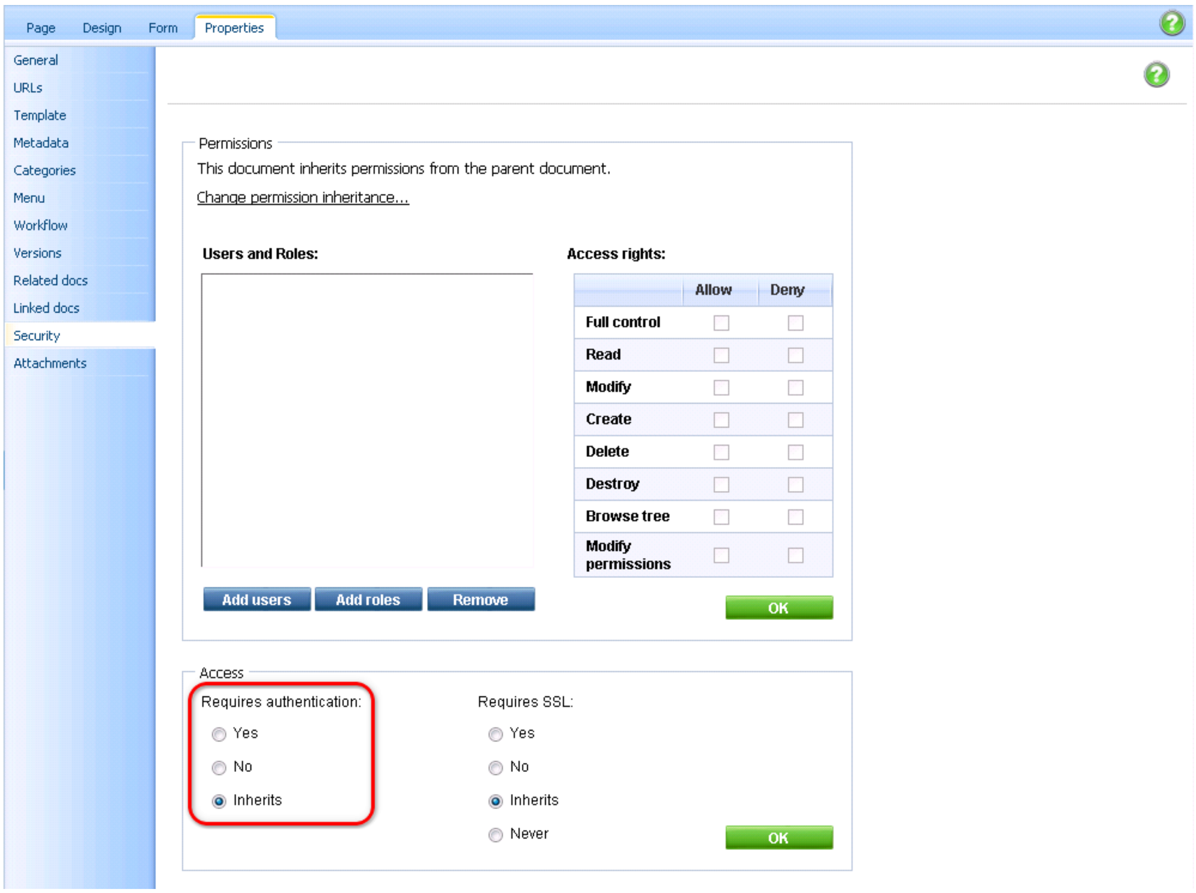
7.10.6 Security

7.10.6.1 Secured website areas

Kentico CMS allows you to easily create secured website areas that are accessible only by authenticated users. When an non-authenticated (public) user comes to the secured section, they are redirected to the logon page specified for the site at **Site Manager -> Settings -> Security -> Website logon page URL**.

You can mark any section of the website as a secured site area by setting **Properties -> Security -> Requires authentication** to:

- **Yes** - page is secured, authentication is required to access it
- **No** - authentication is not required to access the page
- **Inherits** - value of the setting is required from the parent page



Configuration of a secured site area

This example explains how to secure the Products section in the sample Corporate Site.

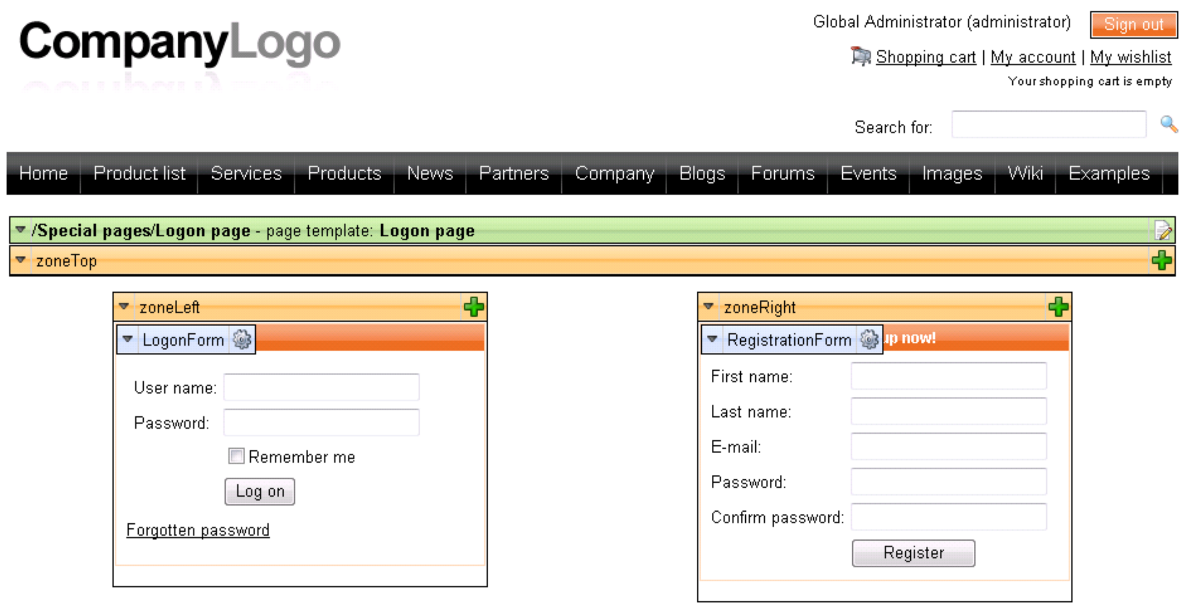
1. Sign in as administrator to **CMS Desk**. Go to the **Content** section and click the **Products** document in the content tree.

2. Click **Properties -> Security**. Set the value of the **Requires authentication** attribute to **Yes** and click **OK**.

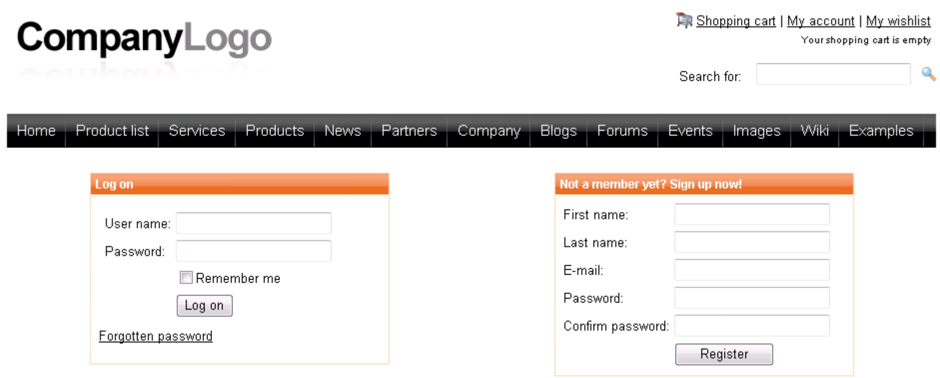
3. Go to **Site Manager -> Settings -> Security** and choose the *Corporate Site* site in the drop-down list. Make sure the **Secured areas logon page** is set to *~/logon.aspx*. This is the URL of the logon page. You can either use the system logon page *~/cmspages/logon.aspx* or you can define your own as it's defined in the Corporate Site sample.

4. Go to **CMS Desk -> Content**, click the *Log-on Page* document and click **Design**. As you can see,

the page is based on page template *Log-on page with registration form* that contains the *Logon form web* part and the *Registration form* web part.



5. Sign out and click **Products** in the main menu. You are redirected to the logon form:



6. Sign in as administrator and you will see the Products section.

Checking access to page content

The page content is not secured by default, even if the current user is denied to read the given page. You need to configure this either by setting **Check permissions** to true in the Editable region web part properties (local configuration) or globally by setting the value in **Site Manager -> Settings -> Security -> Check page permissions** to one of the following values:

- NO - doesn't check any permissions.
- ALL - checks permissions for all pages, including public pages.
- SECUREDAREAS - checks permissions only for pages within a secured area.

When the user is not authorized to read the page, they are displayed with the *Access denied* page. You can configure custom access denied page URL in the **Site Manager -> Settings -> Security -> Access denied page URL** property.

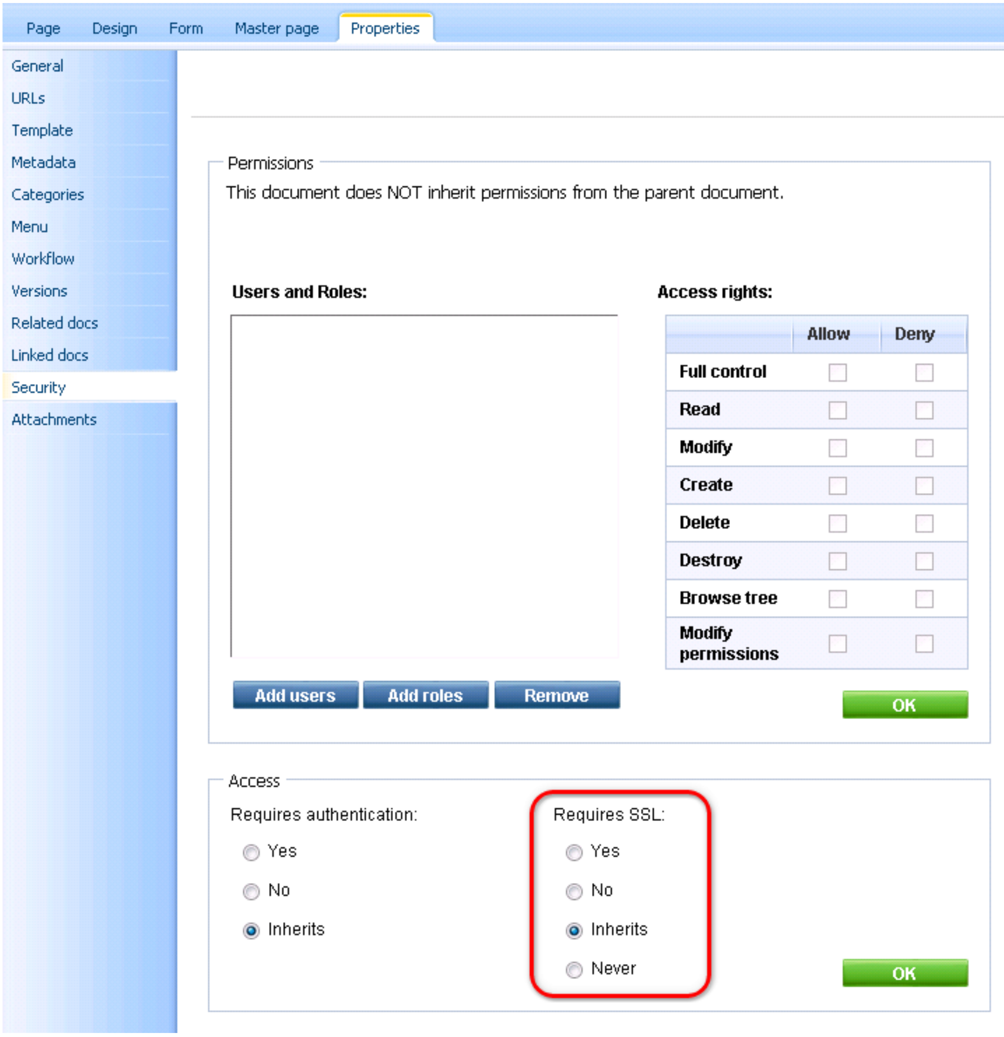
7.10.6.2 SSL (HTTPS) support

Kentico CMS allows you to specify which pages of the website can be displayed only over the secured SSL (HTTPS) protocol. When a user tries to open such a page with standard HTTP protocol, they are redirected to the secured (HTTPS) URL of the same page.

Please note: Kentico CMS doesn't configure your website for SSL/HTTPS, you need to do this manually using standard IIS settings as described in IIS documentation or in this article: http://msdn.microsoft.com/cs-cz/magazine/cc301946(en-us).aspx

In order to specify which pages should be available only through the SSL protocol, you need to go to **CMS Desk -> Content** and select the document in the content tree. Then you can choose one of the following option in **Properties -> Security -> Requires SSL**:

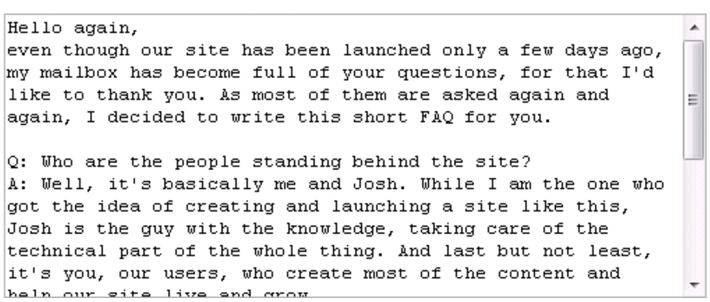
- **Yes** - users trying to access the page will be redirected to the HTTPS URL of the page
- **No** - users trying to access the page will access the page via the protocol from that they are comming (i.e. they will not be explicitly redirected to the secured URL, but if they are coming from a secured URL, they will access the page via secured URL)
- **Inherits** - settings of the parent page will be used
- **Never** - users trying to access the page will be explicitly redirected to the non-secured version of the page



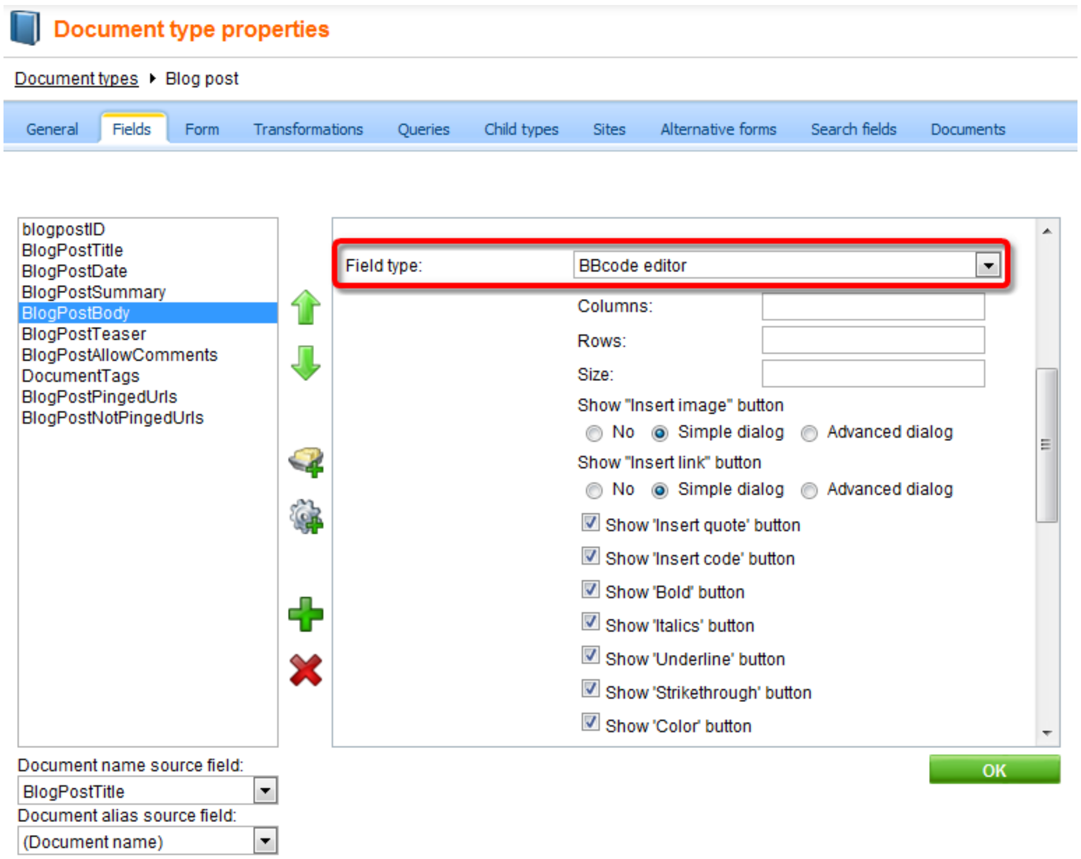
7.10.6.3 Cross site scripting (XSS)

For your websites to be Cross site scripting (XSS) safe, the following rules need to be followed:

1. **Do not use the built-in WYSIWIG editor** (**HTML area (Formatted Text)** field type) to enable users enter text into the site (e.g. user profiles, forums, etc.). Instead use the **BBcode editor**.



This editor is displayed by selecting **BBcode editor** as the value of the Field type attribute when defining document type fields:



2. When writing your transformations, use the following method to resolve text entered via the BBcode editor:

```
CMSHelper.CMSContext.ResolveDiscussionMacros(string inputText)
```

3. When writing your transformations, use the `Eval(string columnName, bool encode)`method with the second parameter enabled to display content of any field whose content was entered by the users, e.g.:

```
<%# Eval("PostSubject",true) %>
```

7.10.6.4 Configuration of allowed request parameters

In some cases, you may need to use super-secure configuration where any non-standard GET or POST parameter sent to your website results in an error. This allows you to avoid some of the possible vulnerabilities, including cross-site scripting and SQL injection.

This functionality is only used for the website, not for the administration interface.

How to configure the allowed parameters

First, you need to enable allowed parameter checking in the web.config file by setting the value CMSCheckParameters to true:

```
<add key="CMSCheckParameters" value="true" />
```

If you're not sure which parameters cause the problem, you can turn on reporting using the following web.config key:

```
<add key="CMSReportCheckParameters" value="true" />
```

All parameters are defined in the **~/parameters.config** file. The schema of the file is described in the file itself and it's rather simple. For every page or site section, you need to define a new <location> section with path specifying the page and allowed form (POST) and query (GET) parameters. The following example allows URL parameter pagenumber in the whole products section of the website:

```
<location path="/products/%">
  <queryparameters>
    <allow param="pagenumber" />
  </queryparameters>
</location>
```

The **path** location specifies the path of the pages based on their alias path in Kentico CMS, while the **page** location is used for single pages that are not part of the Kentico CMS content (custom applications, etc.). The page location starts at the root of the web application and is used without slash (/) at the beginning.

Default allowed parameters

The common parameters of ASP.NET web forms and URL parameters **aliaspath** and **lang** are allowed by default.

7.10.6.5 Managing security using .NET code (API)

You can manage the users, roles and permissions using your own .NET code. Please see the Security Management API examples for more details.

7.10.7 UI personalization

7.10.7.1 Overview

UI personalization enables you to provide certain users of the website with **simplified user interface**. This is useful **typically for business users** who don't need to see all the **tabs**, **menu items**, **actions** and **parts of UI pages** which they don't use. Instead, they have the possibility to see only the things that they need for their job and are not overwhelmed by loads of other options.

Setting up personalized UI can significantly decrease the learning time for new end-users and makes the system generally easier to use and understand for them.

If you are new to UI personalization in Kentico CMS, we recommend you to take the following steps to fully understand it:

1. For UI personalization to be functional, you first need to enable it as described here.

2. It is a good idea to start with the Quick example topic, where you can instantly see what UI personalization is good for.

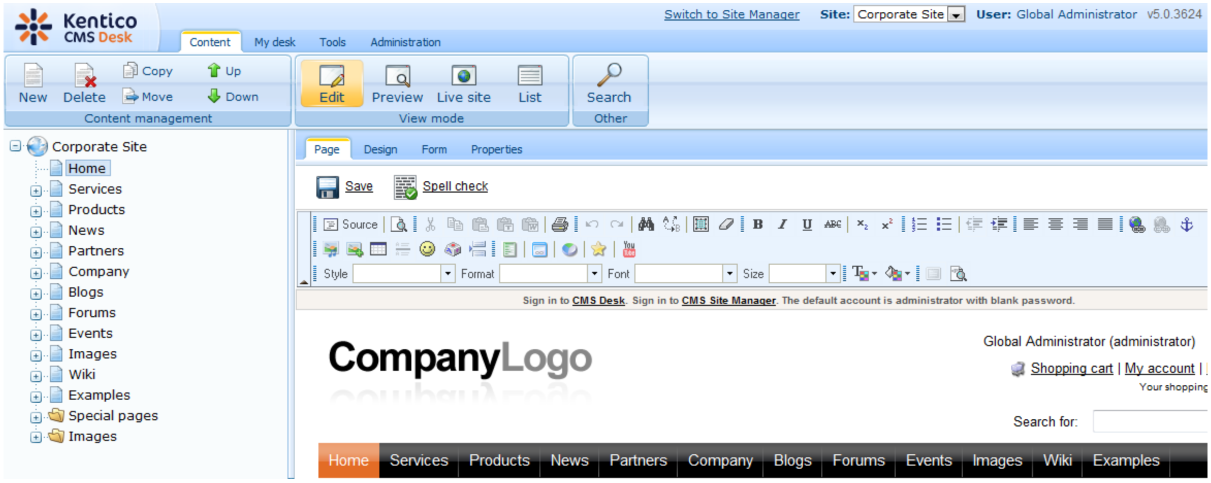
3. With some basic information from the Quick example, you can get deeper knowledge of the terminology, concept and internals in the How does it work chapter.

4. Finally, you can learn what parts of CMS Desk can be hidden in Personalizable parts of CMS Desk.

With the knowledge gained in the chapters above, you can start making UI personalization settings for particular roles as described in Configuring UI personalization for a role. You can also modify or even create your custom UI elements as described in UI elements management and the related example.

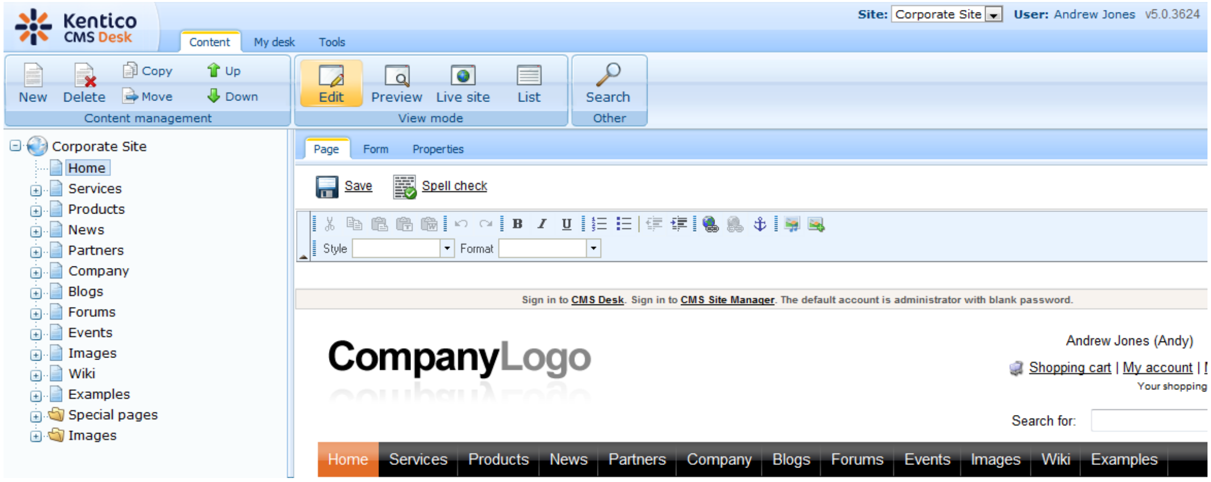
7.10.7.2 Quick example

To see UI personalization in practice, first enable UI personalization as described here. Install one of the sample sites (e.g. Corporate Site) and try logging in to CMS Desk as the **Global Administrator** (login *administrator* with blank password). You will see the full-featured user interface as in the screenshot below.

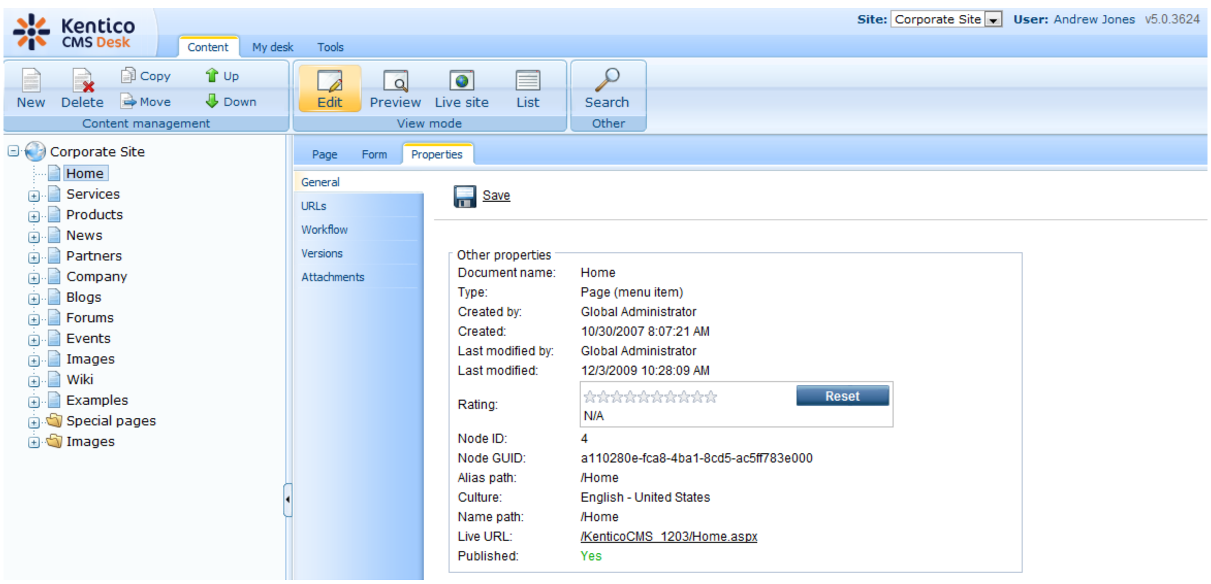


Now log out and try logging in as **Andrew Jones** (login *andy* with blank password). Andy is a member of the **CMS Basic users** role only. This is a role which was added to Kentico CMS to demonstrate the capabilities of UI personalization and its members see only simplified user interface.

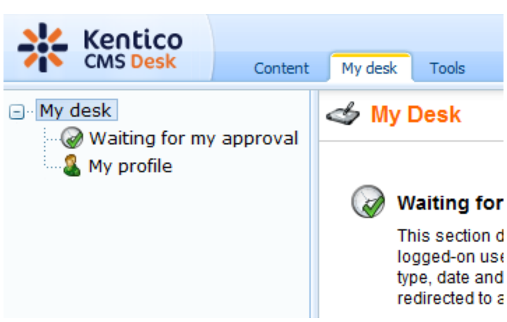
As you can see in the screenshot below, the **Administration** tab in the top menu is not present. The **Design** tab in **Edit** mode is missing too. You can also notice the **WYSIWYG editor**, which offers only a limited number of actions.



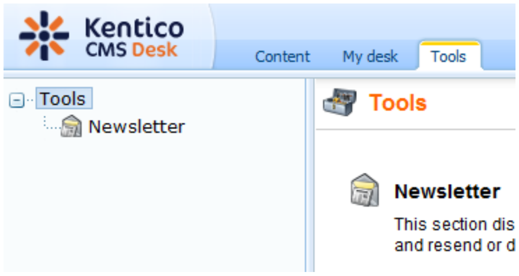
If you switch to the **Properties** tab, you can notice the limited number of items in the left menu. Particular sections accessible via the menu also do not contain all the options available for administrators.



The **My desk** tab contains only two items.



The **Tools** tabs offers only the **Newsletters** module.



It is obvious that this UI is much easier to understand for an end user who only needs to send out newsletters and does not need to do anything else with the CMS.

To see a full overview of which parts of the UI can be hidden, please see Personalizable parts of CMS Desk.

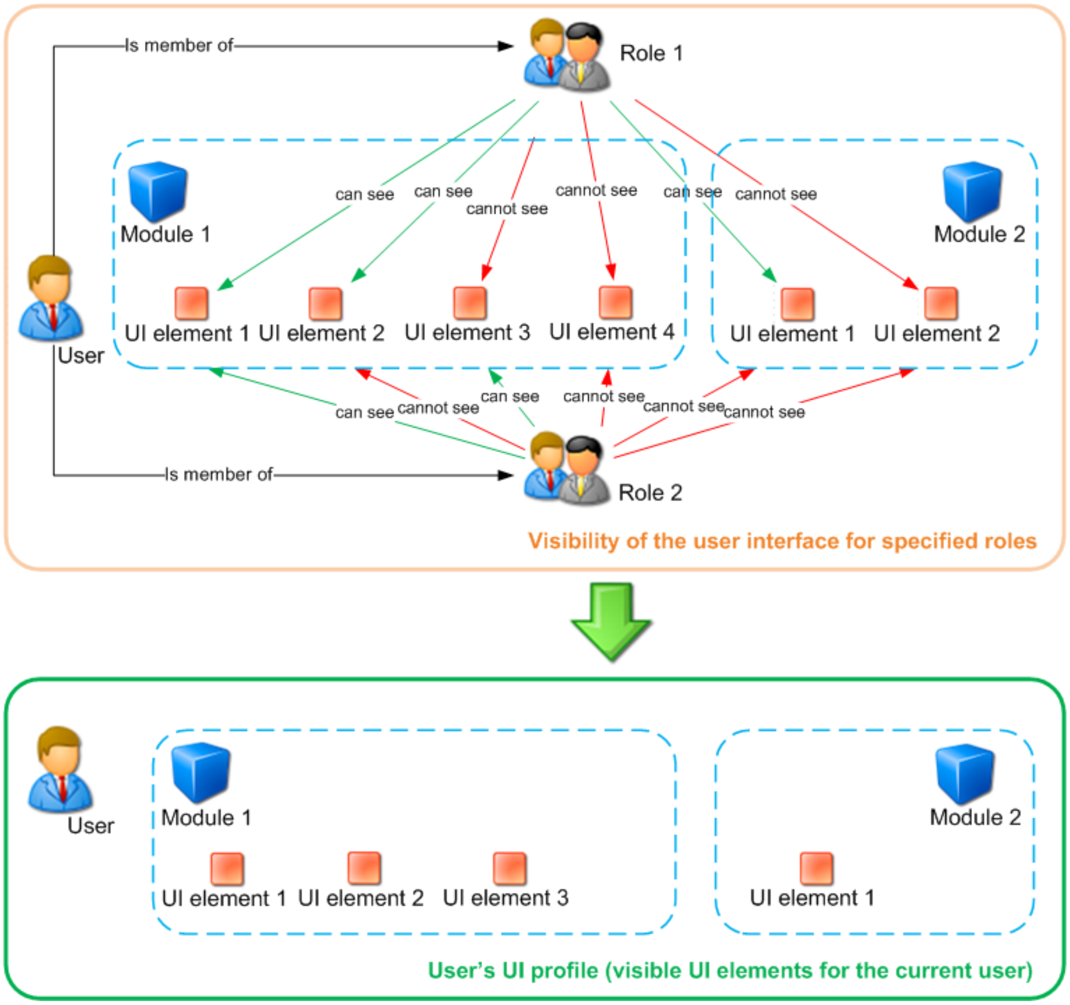
7.10.7.3 How does it work

This chapter explains the principles behind Kentico CMS UI personalization. It is important to get familiar with the following two terms in order to fully understand the concept:

- **UI element** - page or part of a page in CMS Desk which can be hidden from a specified user; it can be a **tab**, a **menu item** or a **group of controls**
- **UI profile** - visibility of specified UI elements for a particular role; it is defined as a set of (role) x (UI element) relationships

Each user's UI profile is defined by UI profiles of their roles. If a user wants to see a UI element, at least one role where the user is member needs to have the UI element set as visible.

The following diagram illustrates how UI personalization settings from two roles are merged to create a user's UI profile.

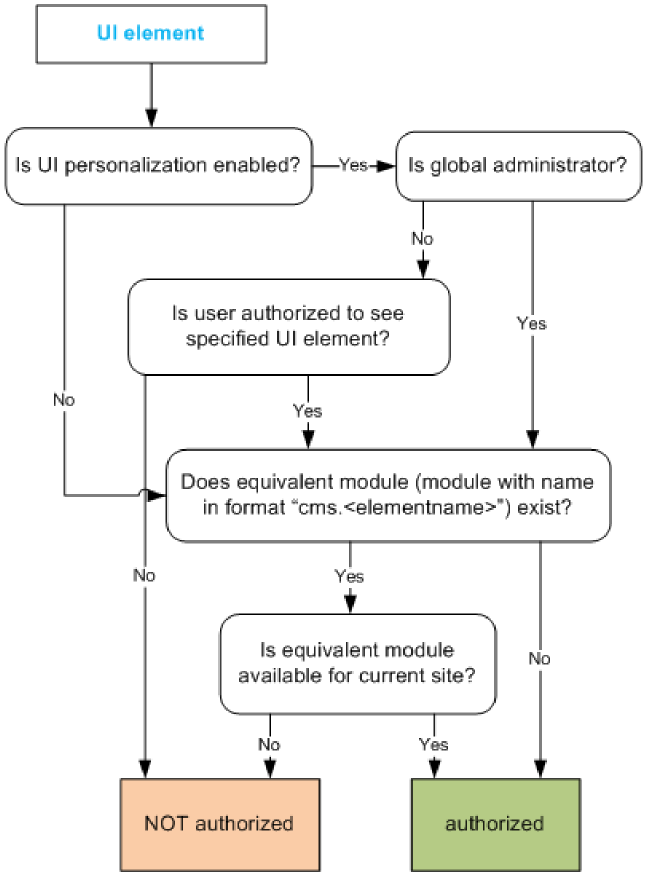


Please note: User's UI profile is **site-related**. This means that the user can see some personalized UI when editing one site and a completely different UI when editing another site. This is possible because a user's UI profile is dependent on the roles. Roles are site-related and a user can be a member of any number of roles across all sites in the system.

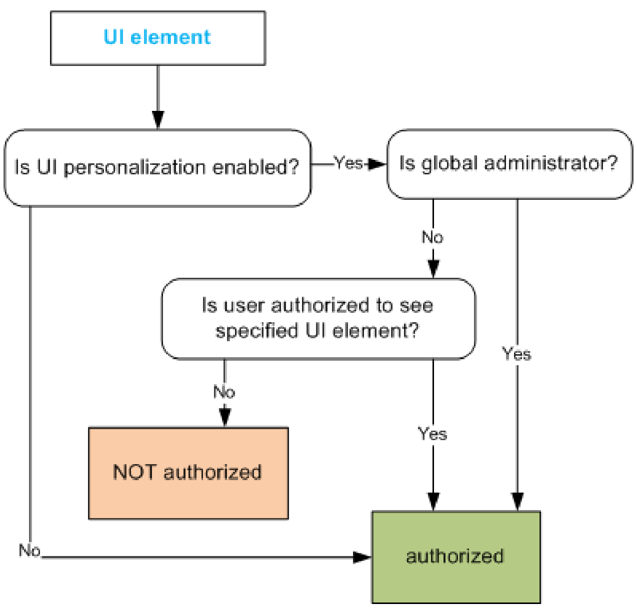
UI element display authorization

The diagram below shows the process of authorization when deciding if a single UI element should be displayed to a user. This process is applied to UI elements in the following parts of the UI:

- main tabs in CMS Desk (Content, MyDesk, Tools, Administration - only the tabs, not their content!)
- left menu in CMS Desk -> MyDesk
- left menu in CMS Desk -> Tools
- left menu in CMS Desk -> Administration



This diagram shows the same authorization process in the remaining tabs and parts of pages:



UI element definitions vs. UI elements in the real user interface

The relation between the UI element definition in **Site Manager -> Modules -> edit ( ) a module -> User interface** and the UI elements in the real UI is different according to the type of UI element.

There are three basic types of UI elements: **tabs**, **menu items** and **groups of controls**.

Tabs in the following locations are **generated dynamically** from the defined UI elements, which means that when a new UI element is defined, the element gets instantly displayed in the UI:

- main tabs in CMS Desk (Content, My desk, ...)
- tabs in CMS Desk -> Edit (Page, Design, ...)
- vertical tabs in CMS Desk -> Edit -> Properties (General, Metadata, ...)
- tabs in CMS Desk -> My desk -> My profile

Menu items in the following menus are also **generated dynamically**:

- CMS Desk -> My desk -> left menu
- CMS Desk -> Tools -> left menu
- CMS Desk -> Administration -> left menu

Groups of controls on particular UI pages (e.g. the Design, Other properties, Owner and Cache sections in CMS Desk -> Content -> Properties -> General) are **pre-defined** and their UI elements are bound to them. You can not define a new group of controls on a page just by definig a new UI element.

More information related to this topic can be found in Personalizable parts of CMS Desk.

7.10.7.4 Personalizable parts of CMS Desk

7.10.7.4.1 Overview

UI personalization can be applied to **CMS Desk only**. Site Manager cannot be personalized as it is typically used by administrators and developers who need the full rather than a simplified UI.

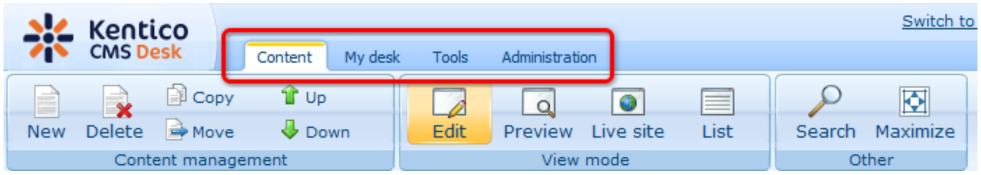
As for CMS Desk, you can personalize the UI elements in the following parts of CMS Desk. Module names in parentheses show which module represents these parts of CMS Desk:

- CMS Desk main tabs (module CMS Desk)
- CMS Desk -> Content tab (module Content)
- CMS Desk -> My desk tab (module My desk)
- CMS Desk -> Tools tab (module Tools)
- CMS Desk -> Administration tab (module Administration)
- WYSIWYG editor (module WYSIWYG editor)
- Media dialog (module Media dialog)

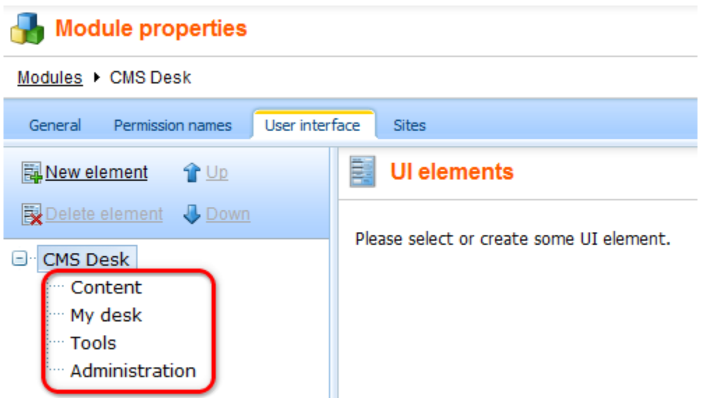
Click particular sections to learn more.

7.10.7.4.2 CMS Desk main tabs

You can hide each of the four main tabs - **Content**, **My desk**, **Tools** and **Administration** - as highlighted in this screenshot:



As you can see, the **CMS Desk** module has UI elements with exactly the same names as the tabs above:

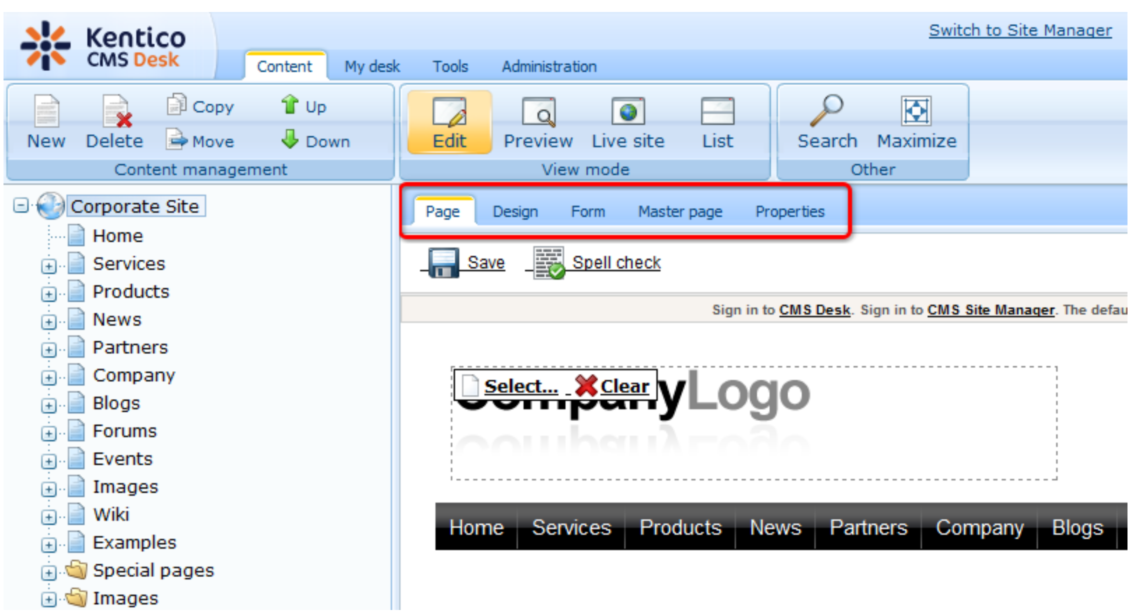


7.10.7.4.3 CMS Desk -> Content tab

The **Content** tab provides vast possibilities of UI personalization. For easier explanation, we will divide them in **three groups**: **Edit mode** tabs, **Design** tab and **Properties** tab.

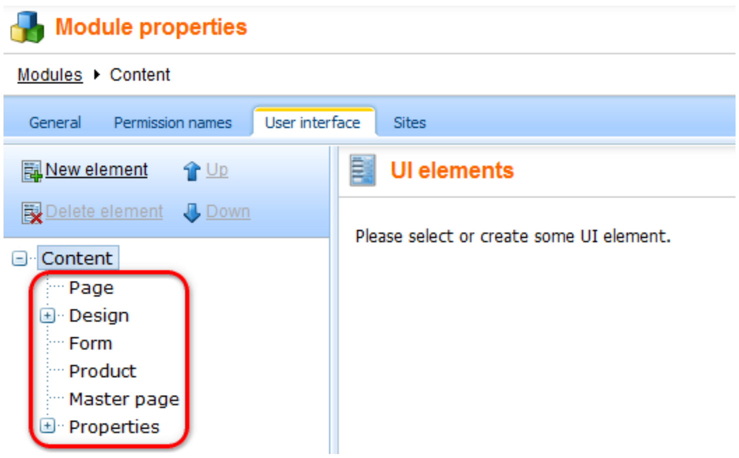
Edit mode tabs

You can hide any of the main tabs displayed in **Edit** mode as highlighted in the screenshot below.



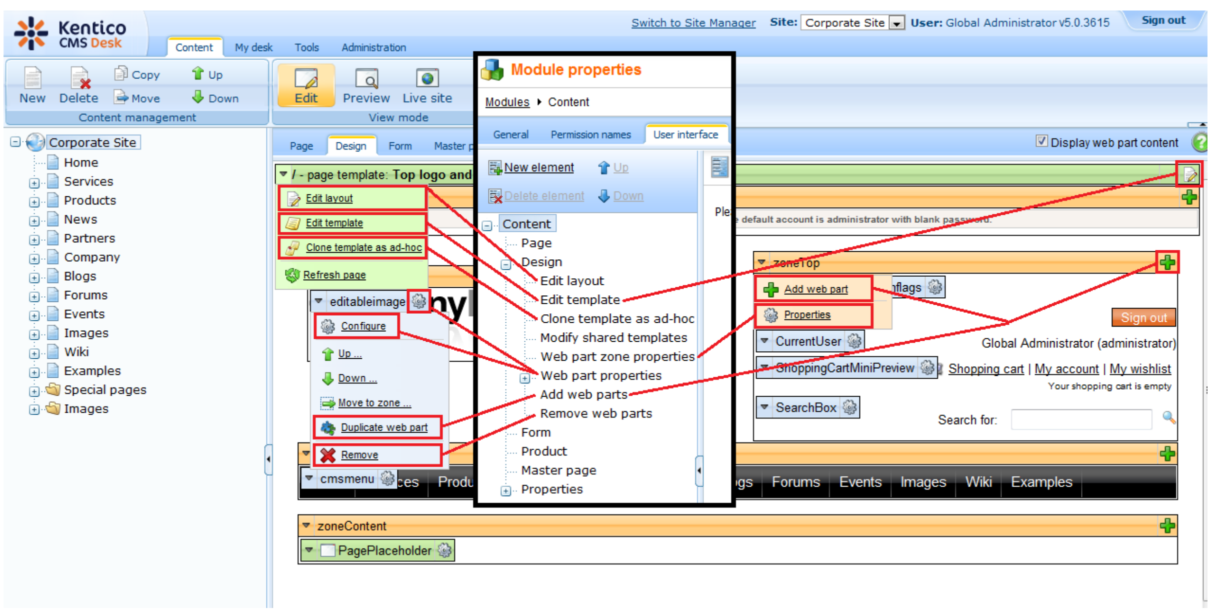
These main tabs are represented by the first-level UI elements of the **Content** module as highlighted below.

Please note that the **Product** tab is not displayed in the screenshot above as it is only displayed when a product type document is selected in the content tree. This is also the case of the **Master page** tab, which is displayed only when you are using Portal engine and the root of the content tree (the master page) is selected.



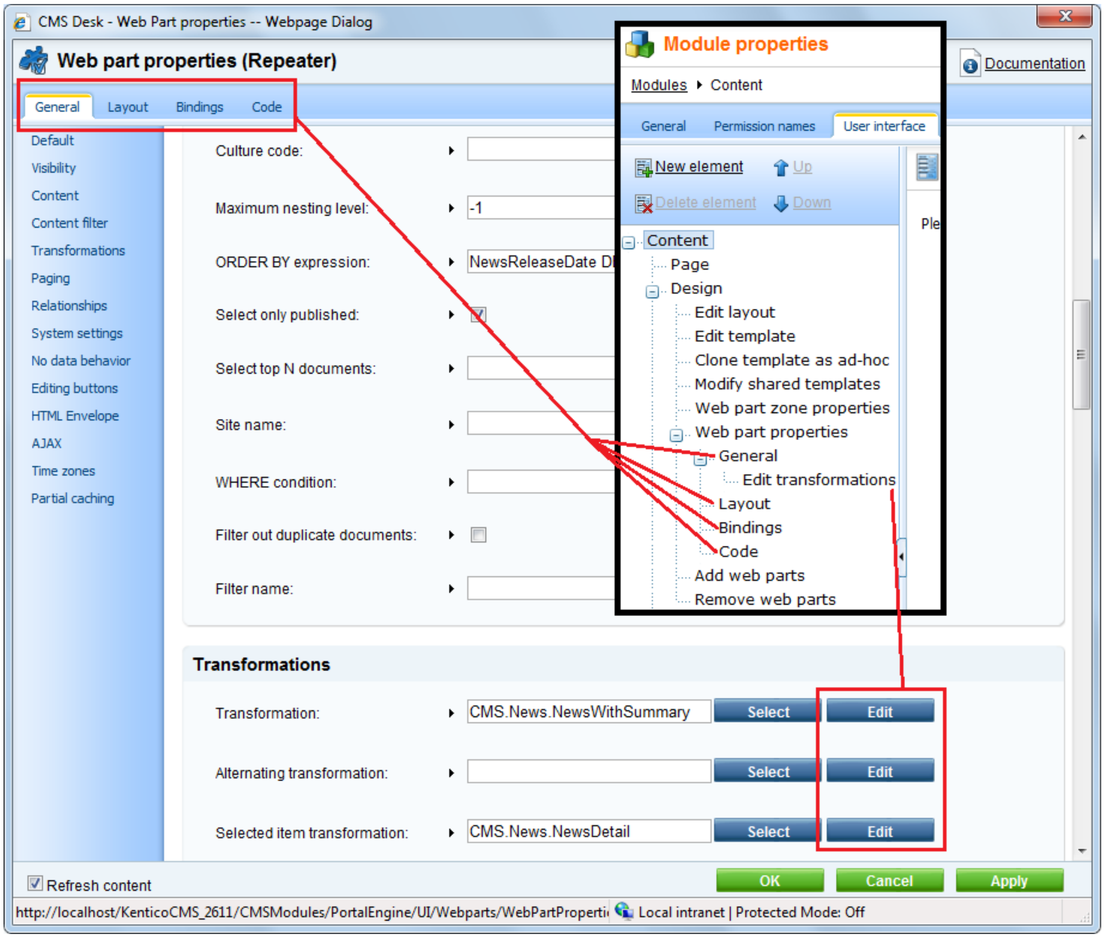
Design tab

On the **Design** tab, you customize all the essential editing actions highlighted in the screenshot below. The corresponding UI elements can be found under the **Design** node of the **Content** module's UI elements tree.



Web part properties dialog

You can hide the four main tabs - **General**, **Layout**, **Bindings** and **Code** - in the **Web part properties** dialog.



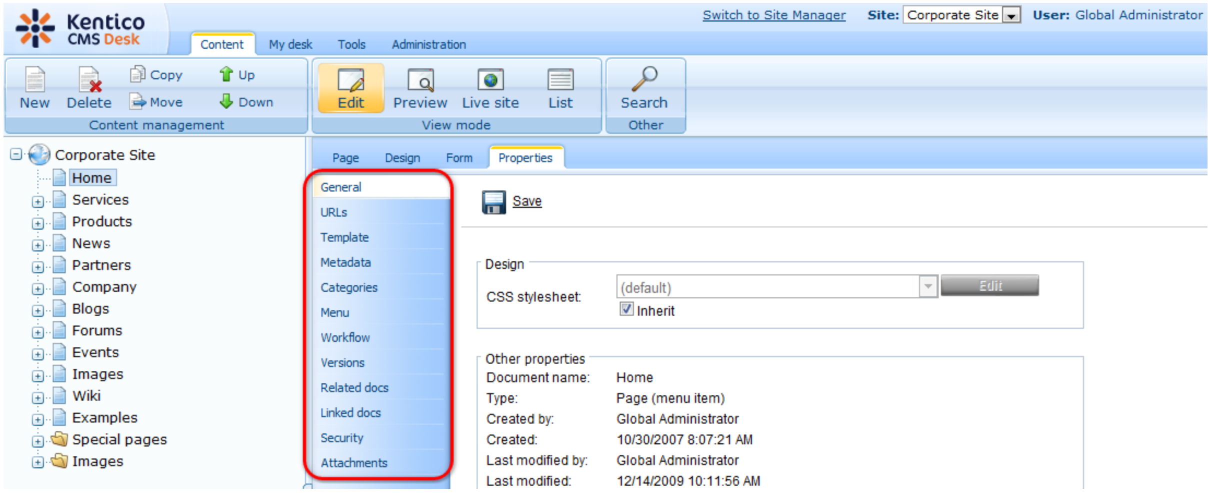
Properties tab

You can hide all items in the left menu on the **Properties** tab. You can also hide particular parts of the **General**, **URLs**, **Template**, **Metadata**, **Menu** and **Security** pages. More information can be found on this page.

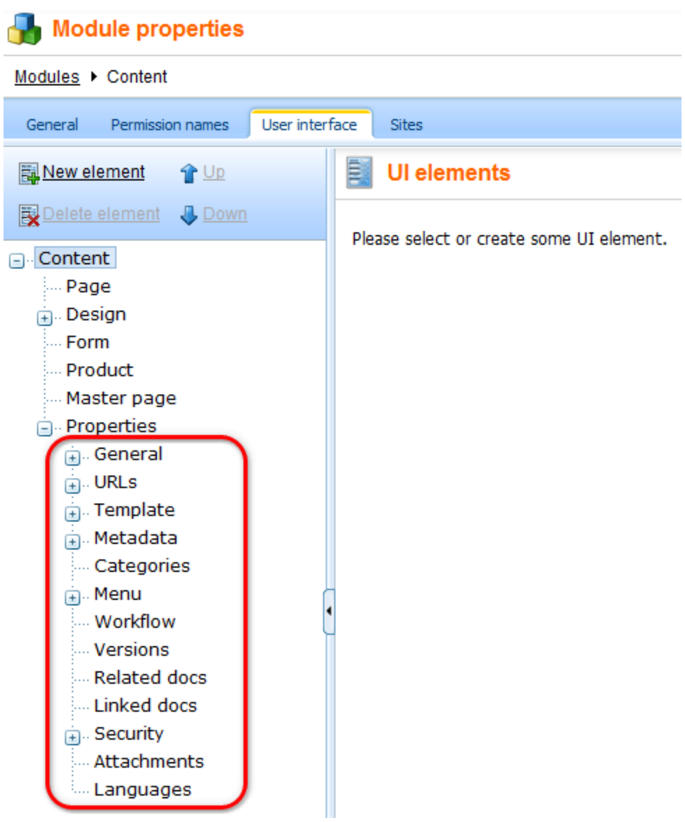
7.10.7.4.4 CMS Desk -> Content -> Edit -> Properites tab

This chapter describes UI personalization possibilities for the **Content -> Edit -> Properties** tab of a document.

First of all, you can hide all items in the left menu on the **Properties** tab as highlighted in the screenshot below.



In the **Content** module, you can find the relevant UI elements under the **Properties** node. These elements have the same names as the names of particular menu items.

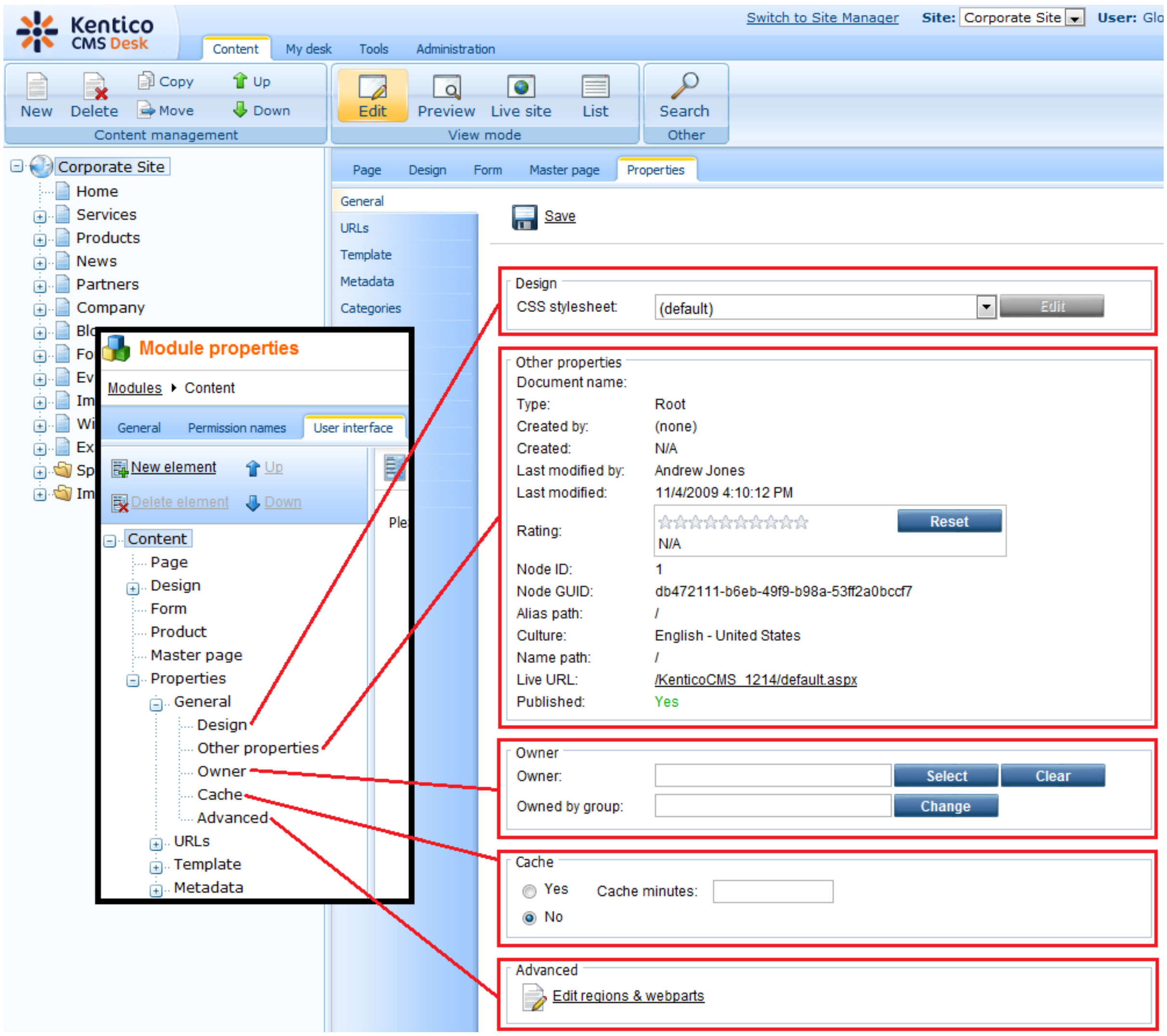


You can also expand the **General**, **URLs**, **Template**, **Metadata**, **Menu** and **Security** UI elements using the ⊞ icon in order to display their child UI elements. These child UI elements represent parts of the relevant pages as described in the text below:

General tab

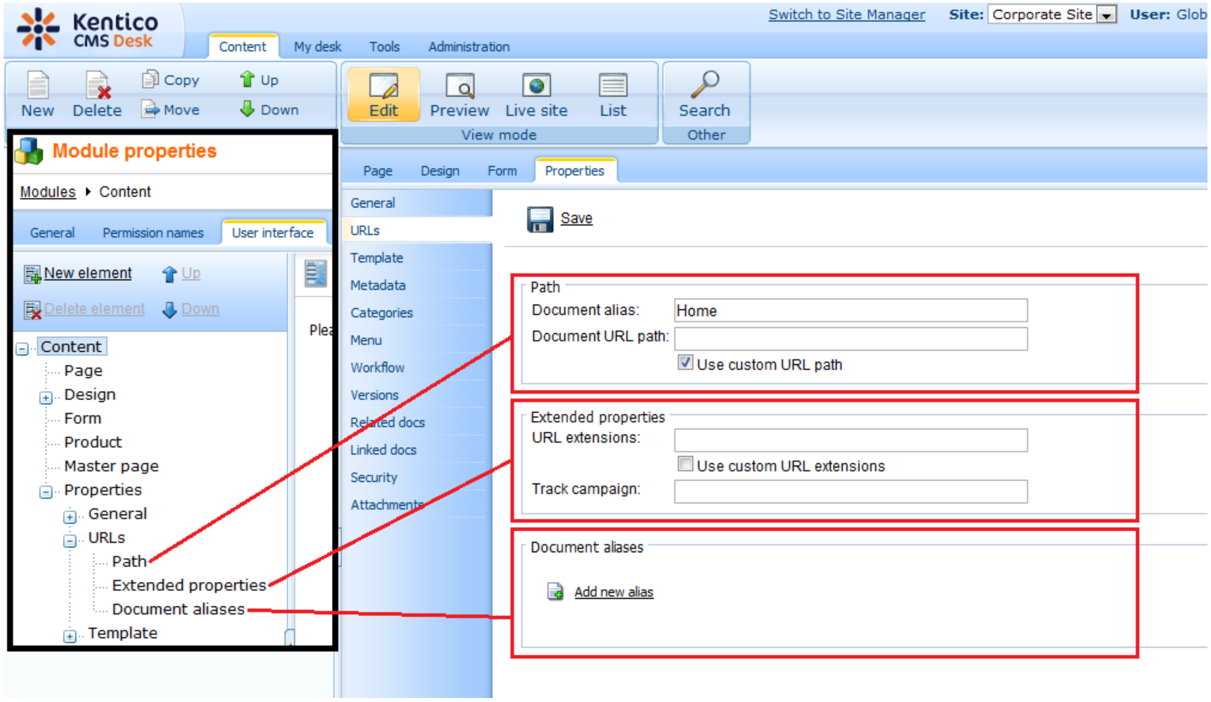
The **General** tab consists of the **Design**, **Other properties**, **Owner**, **Cache** and **Advanced** sections.

The related UI elements have the same names as these sections, as you can see in the following screenshot:



URLs tab

The URLs tab consists of the **Path**, **Extended properties** and **Document aliases** sections. The related UI elements have the same names as these sections, as you can see in the following screenshot:



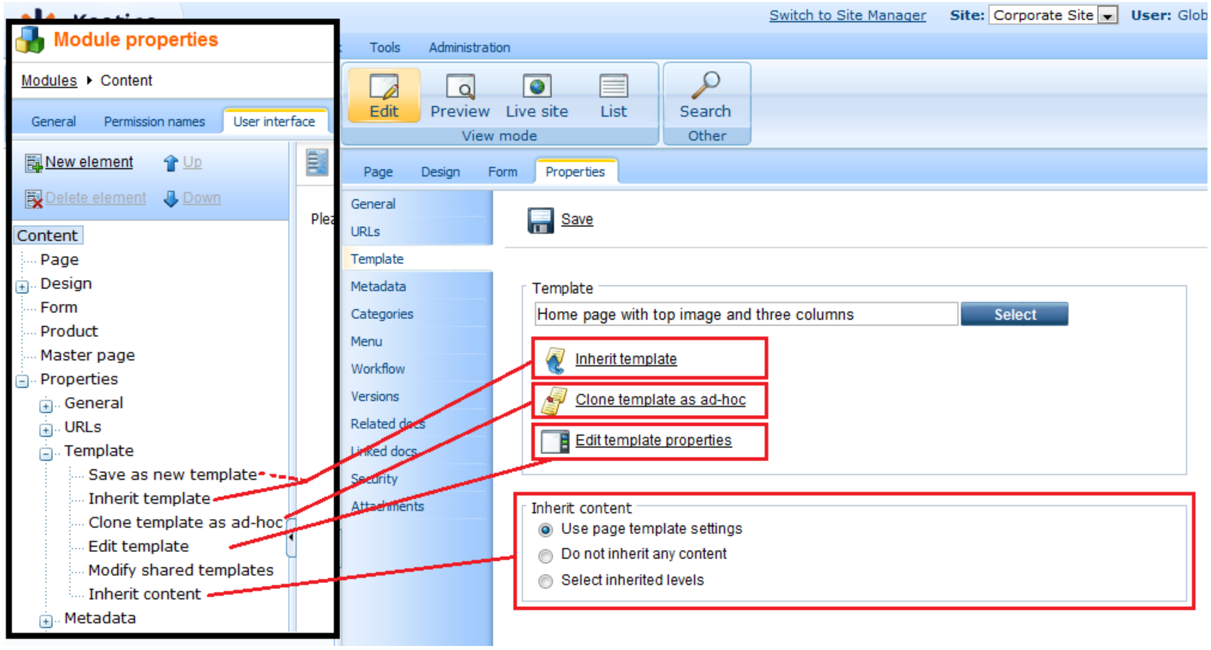
Template tab

On the **Template** tab, you can hide the **Save as new template**, **Inherit template**, **Clone template as ad-hoc** and **Edit template properties** actions. These actions are represented by the UI elements with the same names, as you can see in the screenshot below.

You can also hide the whole **Inherit content** section using the UI element with the same name.

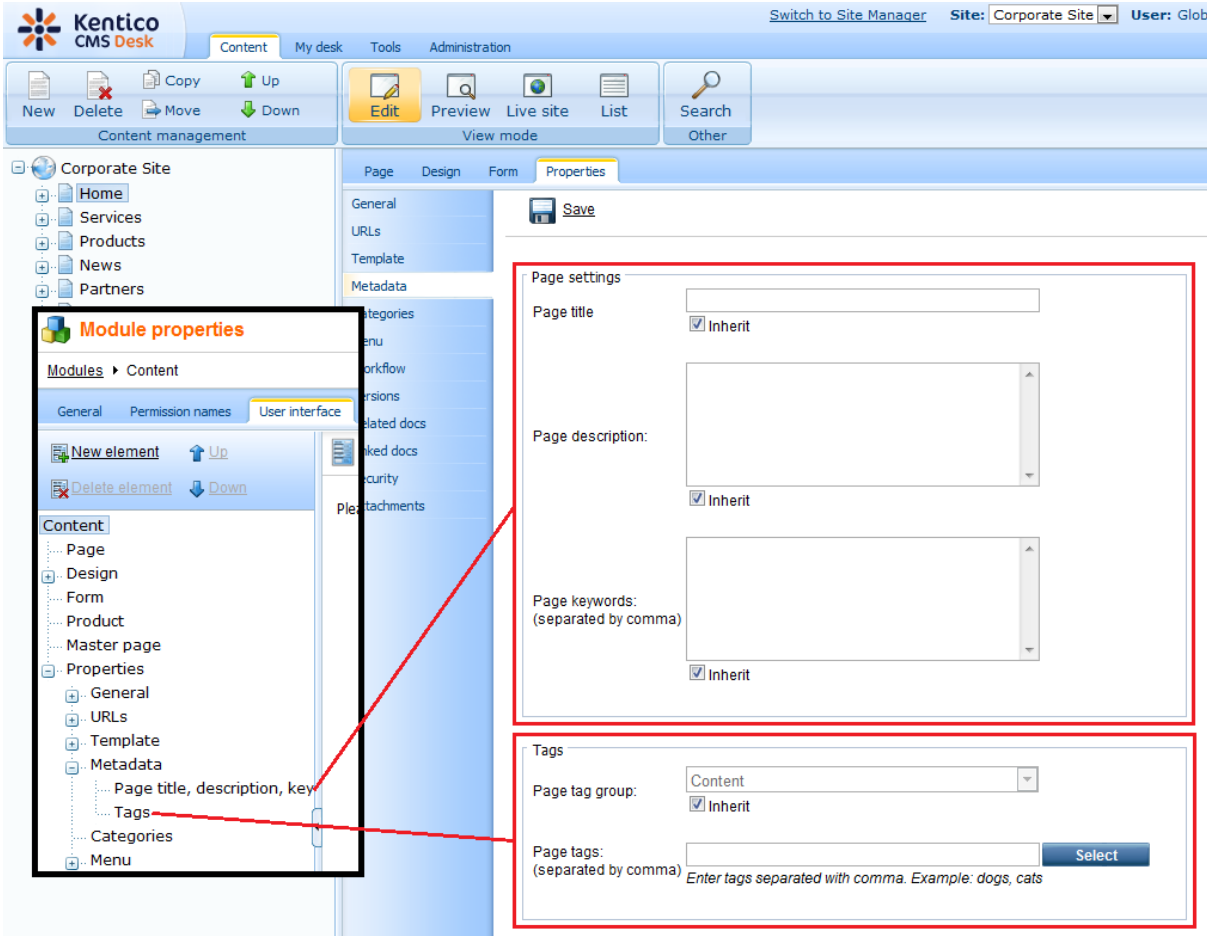
The **Modify shared templates** UI element works as a permission: it determines if the users can modify shared page templates both from here and on the **Design** tab.

Please note that the **Save as new template** action is not displayed in the screenshot because it is available only for ad-hoc or reusable templates. If displayed, the action is positioned instead of the **Inherit template** action. This is why the UI element is connected by a dotted line with the action button.



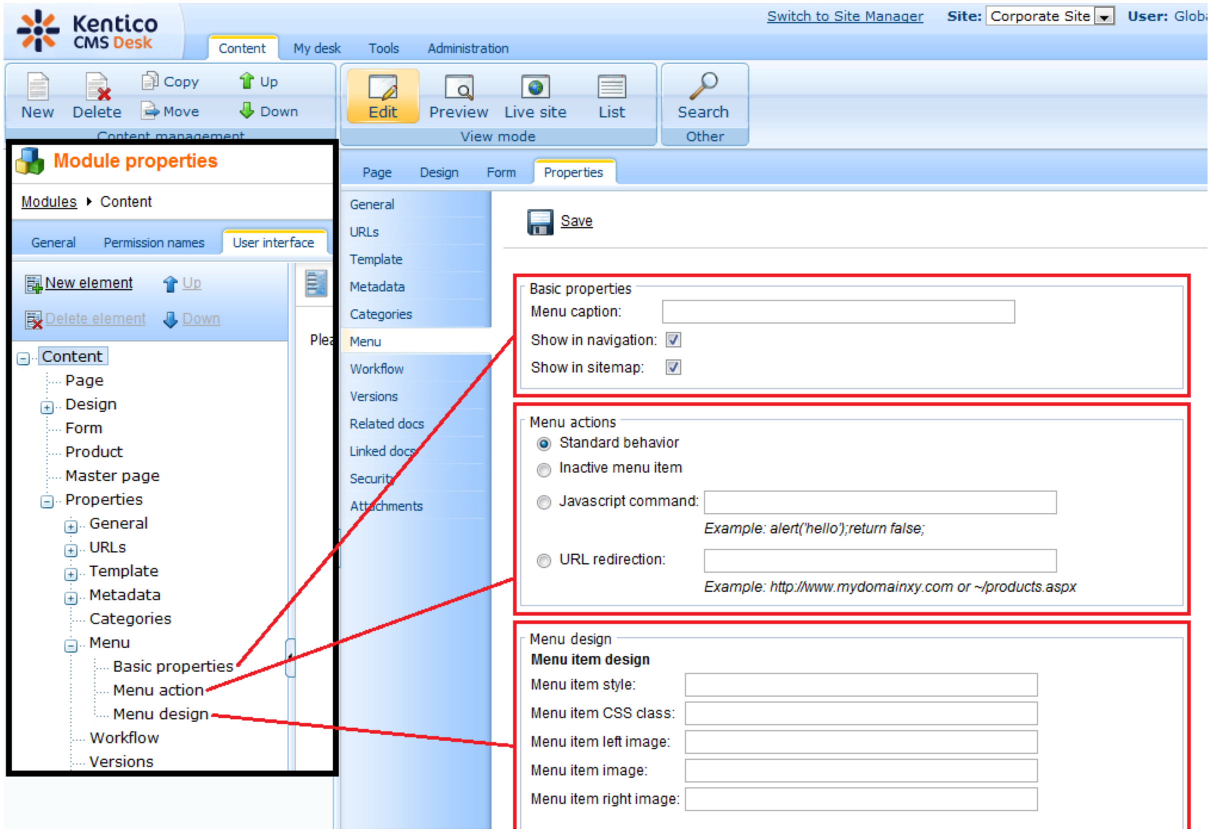
Metadata tab

The Metadata tab consists of two main sections. The **Page settings** section is represented by the **Page title, description, keywords** UI element. The **Tags** section is represented by the **Tags** UI element.



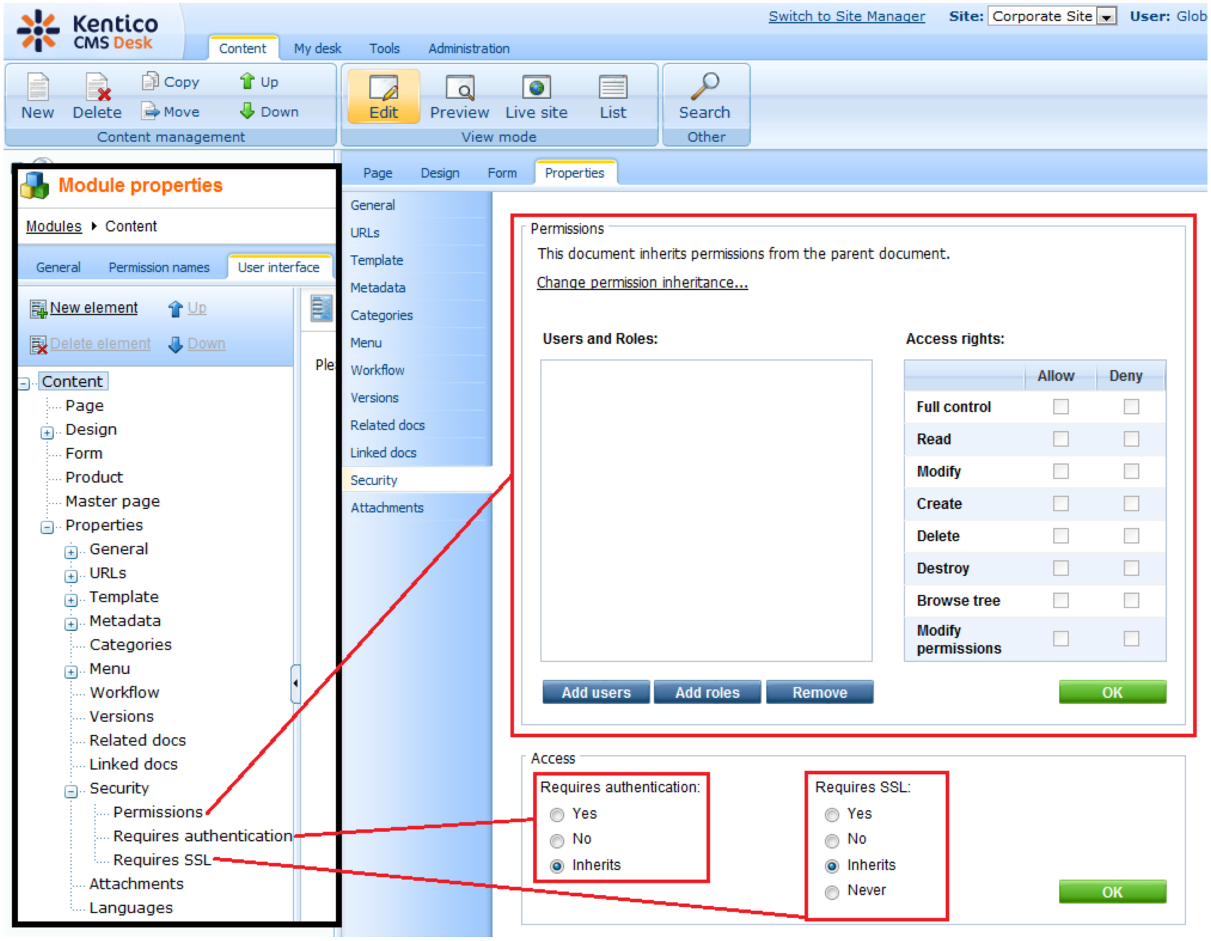
Menu tab

The **Menu** tab consists of three sections - **Basic properties**, **Menu actions**, **Menu design** - while these sections are represented by UI elements with the same names as shown in the following screenshot:



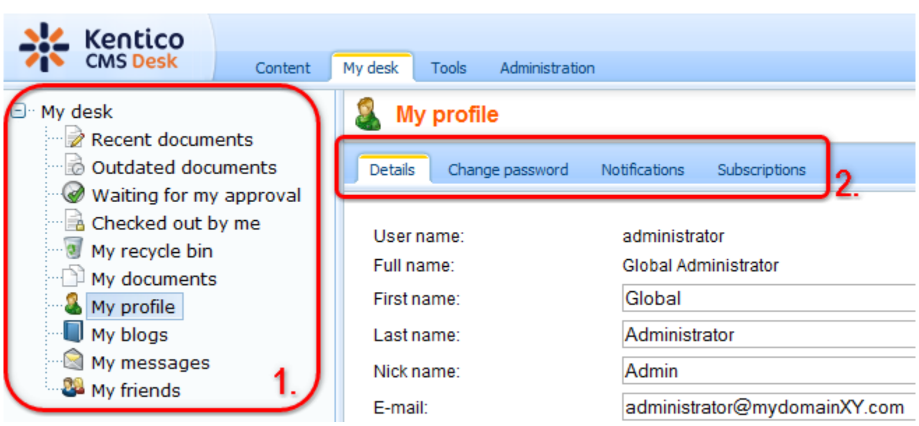
Security tab

The **Security** tab consists of the **Permissions** section, which is represented by the **Permissions** UI element, and the **Access** section, which is divided into two UI element - **Requires authentication** and **Requires SSL**:

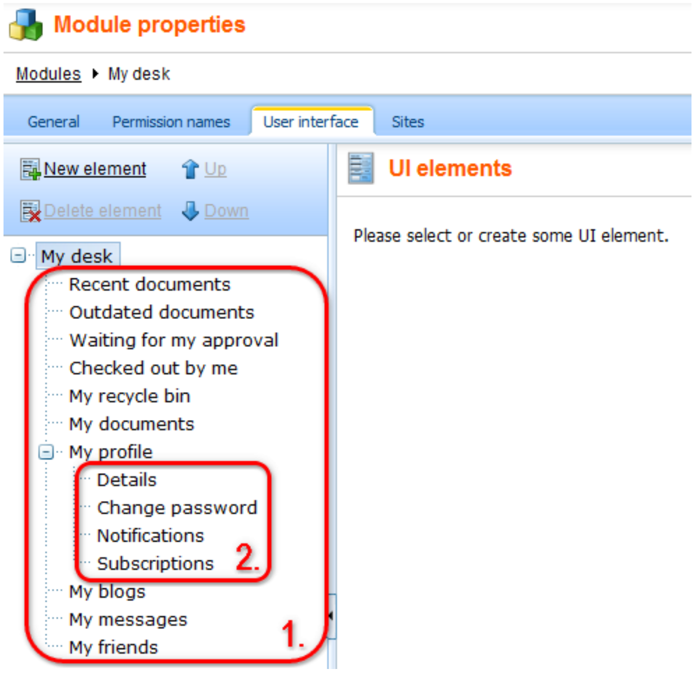


7.10.7.4.5 CMS Desk -> My desk tab

You can hide each item in the left menu (1.). Additionally, you can also hide all four tabs on the **My profile** page (2.).

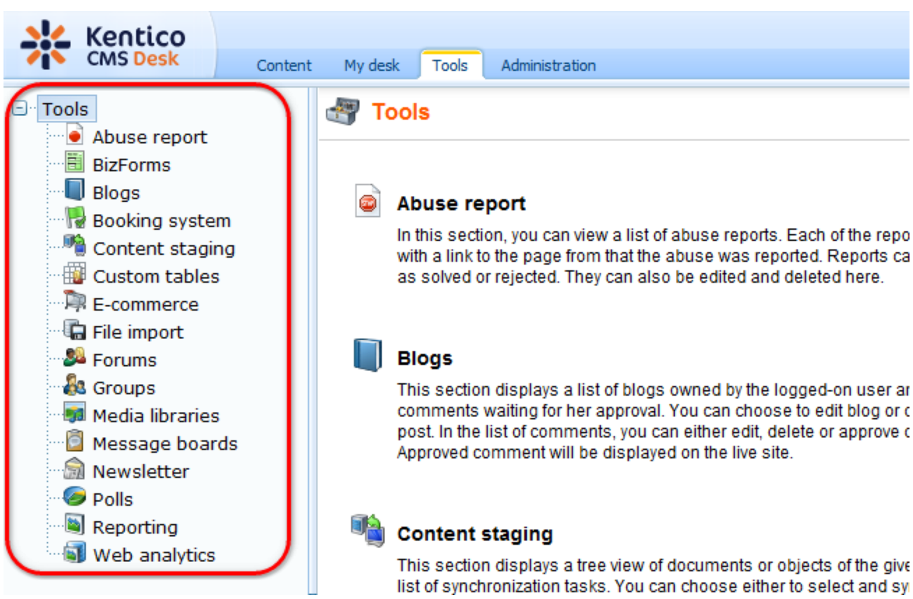


In the screenshot below, you can see UI elements of the **My desk** module. Their names match the names of the menu items and tabs on the screenshot above.

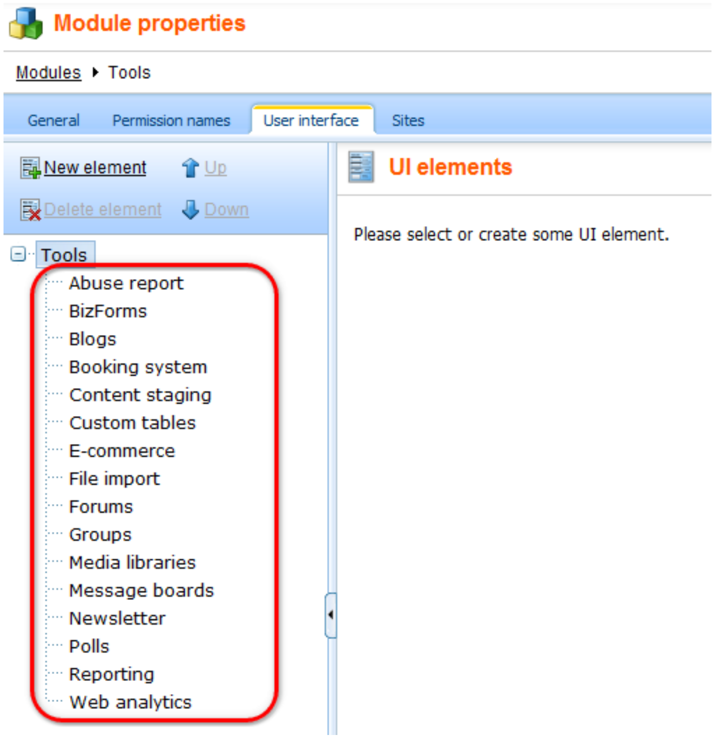


7.10.7.4.6 CMS Desk -> Tools tab

You can hide each item in the left Tools menu as highlighted in the following screenshot:

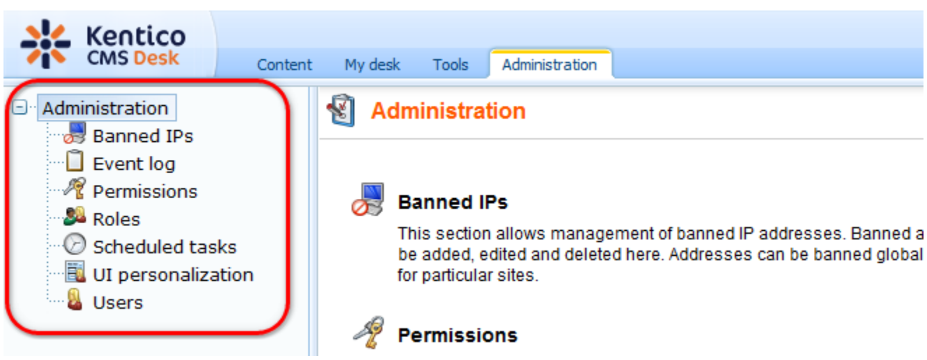


The **Tools** module has UI elements with the same names as the menu items in the screenshot above.

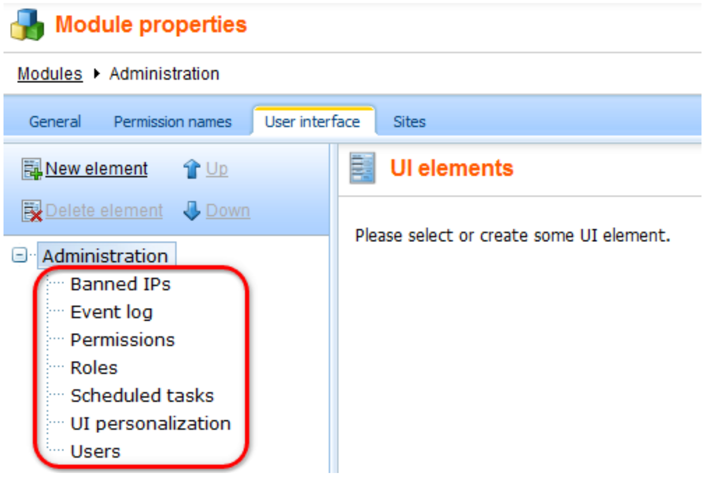


7.10.7.4.7 CMS Desk -> Administration tab

You can hide all items in the left Administration menu.

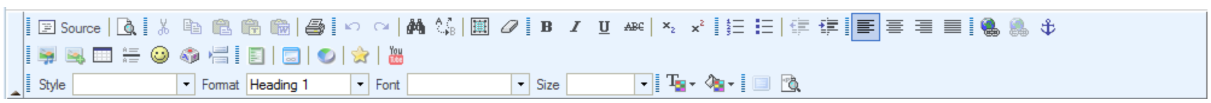


The **Administration** module has UI elements with the same names as the menu items in the screenshot above.



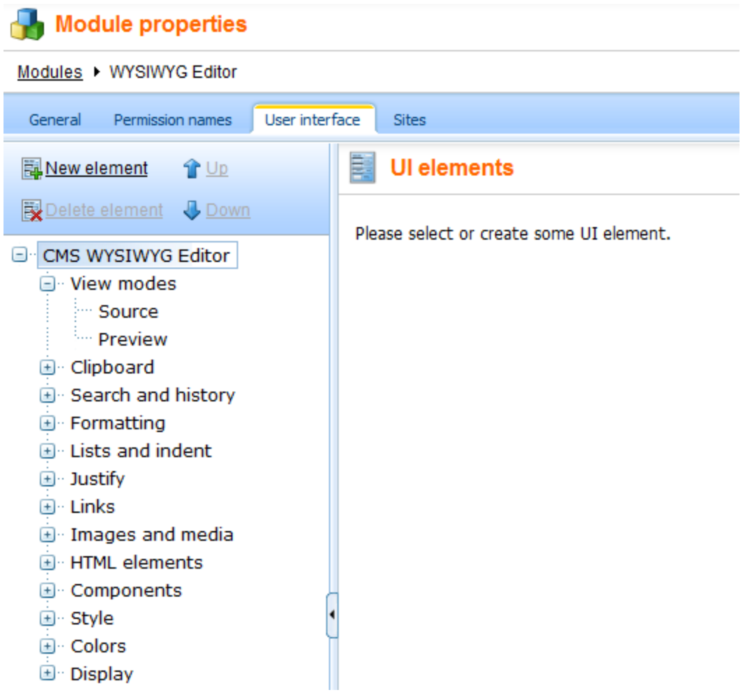
7.10.7.4.8 WYSIWYG editor

All action buttons on the [WYSIWYG editor](#) toolbar can be hidden. This can be particularly useful for content editors who do not need to use all these actions.



The **WYSIWYG editor** module has many UI elements grouped under several parent UI elements. The parent elements serve only for categorization purposes and do not represent any action buttons in the toolbar. You can display the sub-elements by clicking the ⊞ icon next to a particular UI element.

UI element names match tooltips displayed when you place your cursor over a particular button in the WYSIWYG editor toolbar.



Custom toolbar actions

If you use your custom plug-in in the WYSIWYG editor and want its action button to be displayed or hidden based on UI profile settings, you need to create a new UI element in the WYSIWYG Editor module with the same code name as the code name of the plug-in.

How it works

1. Action buttons are loaded into the toolbar based on the toolbar definition.
2. If UI personalization is enabled, action buttons get filtered according to the user's UI profile.

This means that only such UI elements are displayed in the toolbar that are available both in the user's UI profile and in the toolbar definition.

Toolbar personalization on the live site

Toolbar personalization on the live site is disabled by default. To change this, you need to enable it in your web.config file by adding the following key to the **/configuration/appSettings** section:

```
<add key="FCKEditor:PersonalizeToolbarOnLiveSite" value="true" />
```

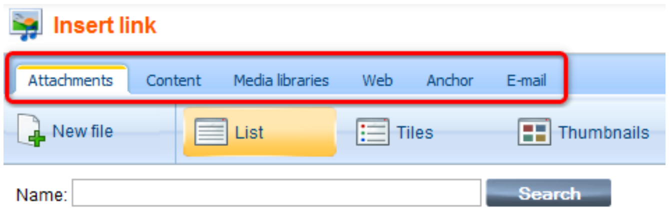
Once this is done, all UI personalization settings will be applied to the WYSIWYG editor on the live site.

More web.config settings can be found in Appendix C - Web.config parameters -> WYSIWYG editor settings.

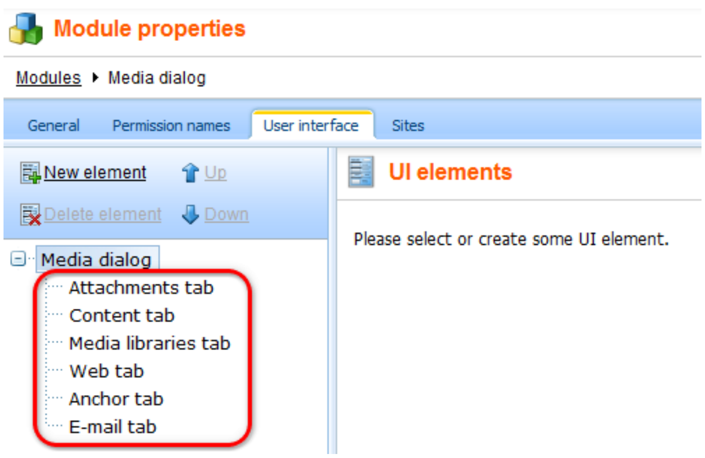
7.10.7.4.9 Media dialog

This dialog is displayed e.g. when using the Insert image or media or Insert link actions in the WYSIWYG editor or when selecting an image in the **Editable image** web part.

You can hide the dialog's tabs - **Attachments**, **Content**, **Media libraries**, **Web**, **Anchor** and **E-mail**.



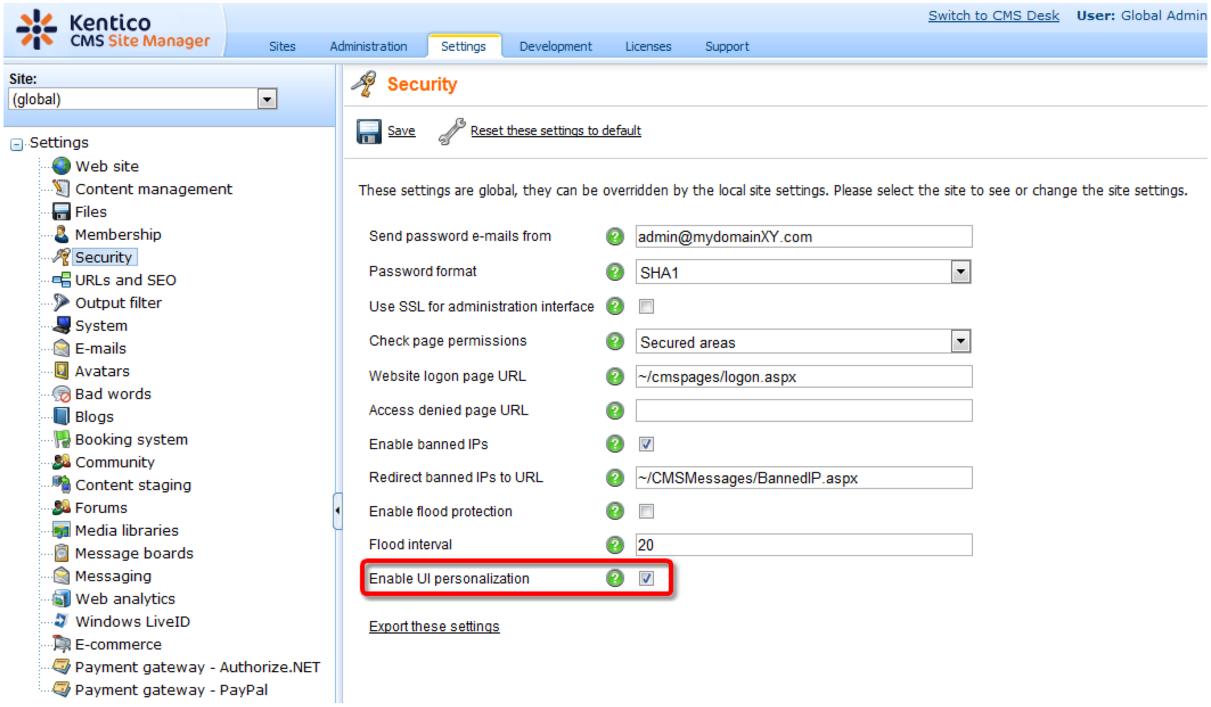
UI elements are defined centrally in the **Media dialog** module. They are applied to all occurrences and versions of the dialog, even though some of these tabs are not displayed in all cases (e.g. the E-mail tab is not displayed in Insert image or media dialog).



7.10.7.5 Enabling UI personalization

For UI personalization to be functional, you need to go to **Site Manager -> Settings -> Security** and enable it using the **Enable UI personalization** check-box.

Using the **Site** drop-down, you can decide if you want to turn it on globally, or just for some particular sites in the system.



7.10.7.6 UI profile configuration

In the following step-by-step example, you will learn how to define UI profile for a new role in **Site Manager -> Administration -> UI personalization**.

Let's presume that we want some users to be **Forum administrators** of our website. This means that they will not need to use the **Content** or **Administration** tabs at all, all they will be concerned about is the Forums module administration interface in the **Tools** menu and a few parts of the **My desk** section.

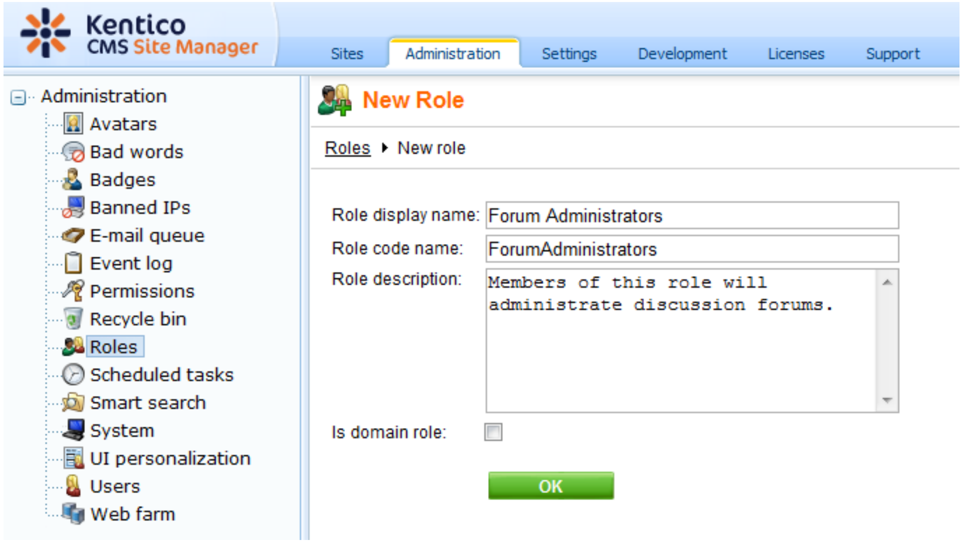
How to achieve it? We will create a new role and a new user who will be a member of this role only. Then we will define UI profile settings for this role and try logging in as the new user to see the simplified UI.

1. Install some of the sample sites or your own site and enable UI personalization for the site as described here.

2. Go to **Site Manager -> Administration -> Roles**, choose your site in the **Site** drop-down and click the **New role** ( ) icon. In the following dialog, enter these details:

- **Role display name**: Forum Administrators
- **Role code name**: ForumAdministrators
- **Role description**: some text describing the role
- **Is domain role**: leave the field blank

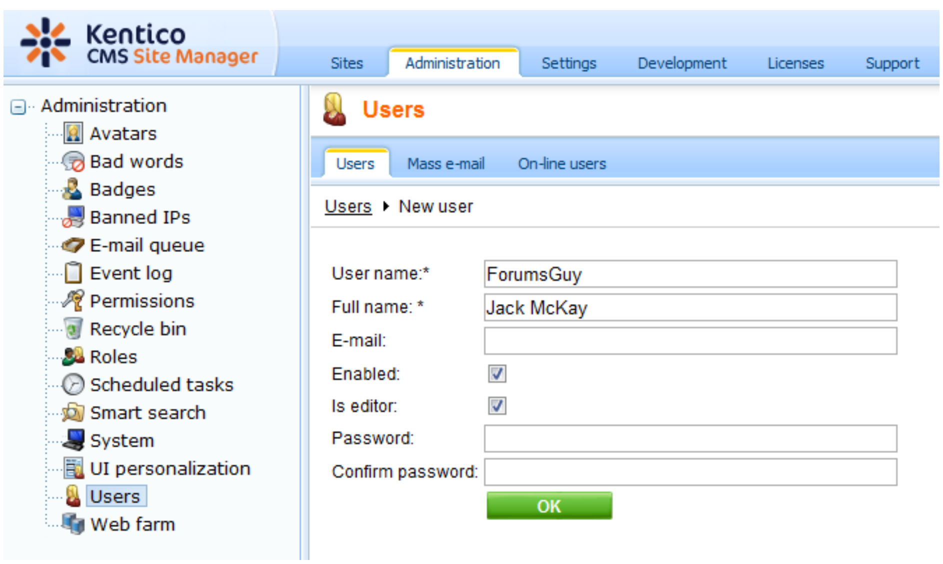
Click **OK**.



3. Now let's create the user. Go to **Site Manager -> Administration -> Users** and click the **New user** ( ) icon. Enter the following details in the New user dialog:

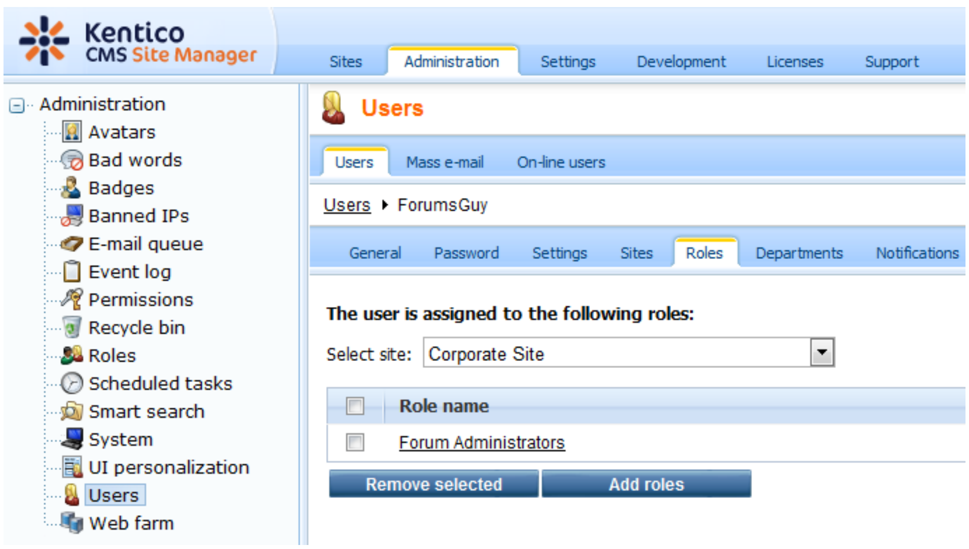
- **User name**: ForumsGuy
- **Full name**: Jack McKay
- **Enabled**: enabled
- **Is editor**: enabled

Click **OK**.



4. You are redirected to the new user's editing interface. We first need to assign the user to our site. Go to the **Sites** tab and add the site using the **Add sites** button.

5. Now that the user belongs to our site, we can assign her to the **Forum Administrators** role, which also belongs to the site. Switch to the **Roles** tab and use the **Add role** button to add the Forum Administrators role.



6. Now that the user belongs to our role, we can set up UI personalization for the role. Go to **Site Manager -> Administration -> UI personalization** and chose:

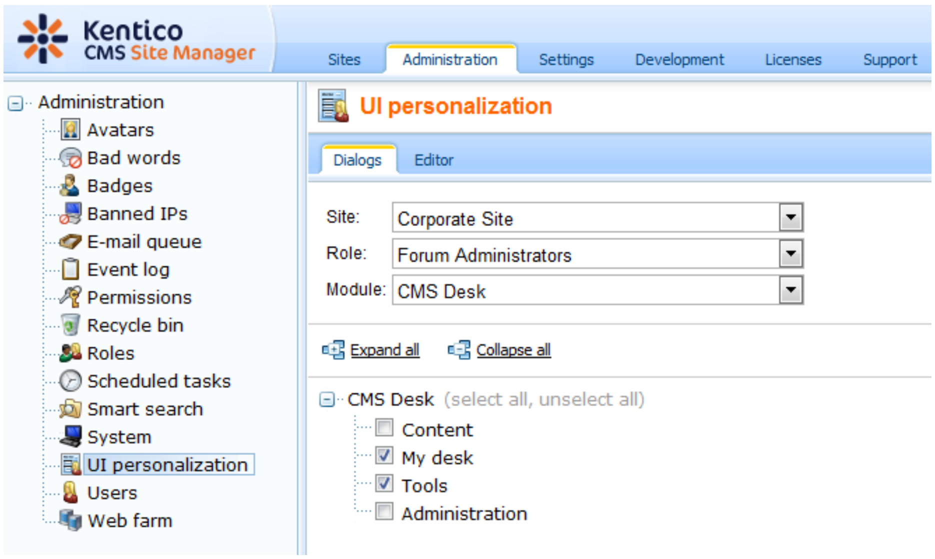
- **Site**: your site
- **Role**: Forum Administrators

Please note: UI personalization settings are **site-related**. This means that members of one role can see some personalized UI when editing one site and a completely different UI when editing another site. More information on how UI personalization works can be found here.

You can choose the module whose UI elements you want to set up using the **Module** drop-down. Full reference on the personalizable parts of CMS Desk and the appropriate modules can be found here.

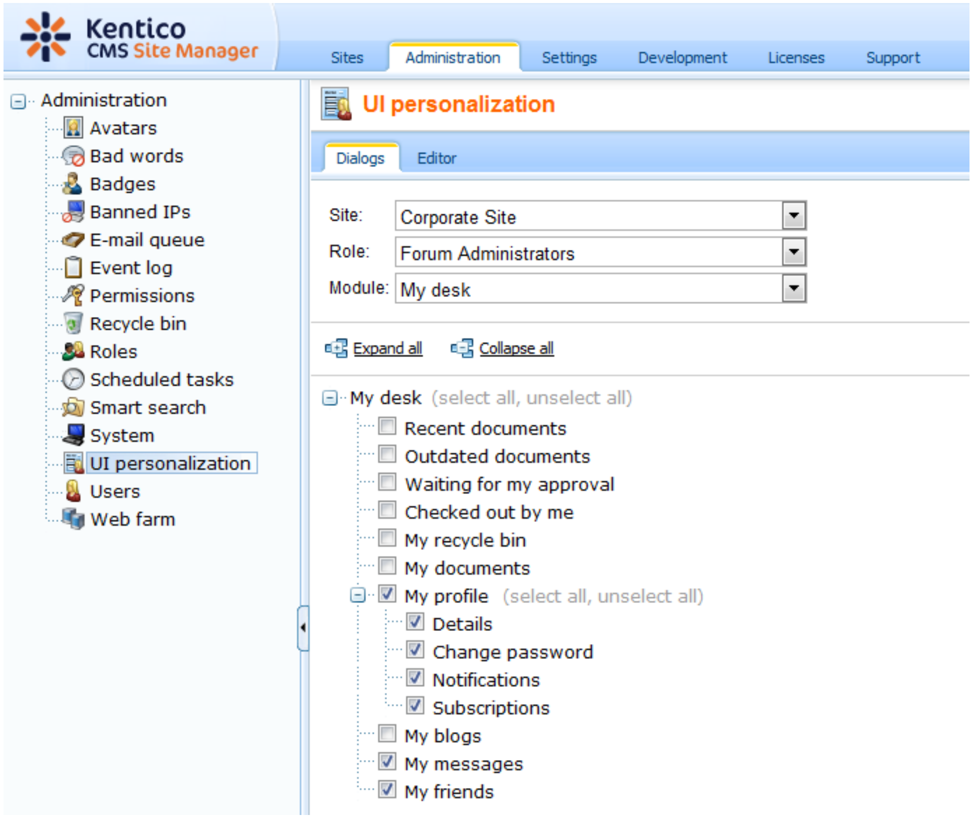
7. Let's start with the main tabs in CMS Desk, where we want only the **My desk** and **Tools** tabs visible. Choose **Module**: CMS Desk and make the following settings:

- **Content**: disabled
- **My desk**: enabled
- **Tools**: enabled
- **Administration**: disabled

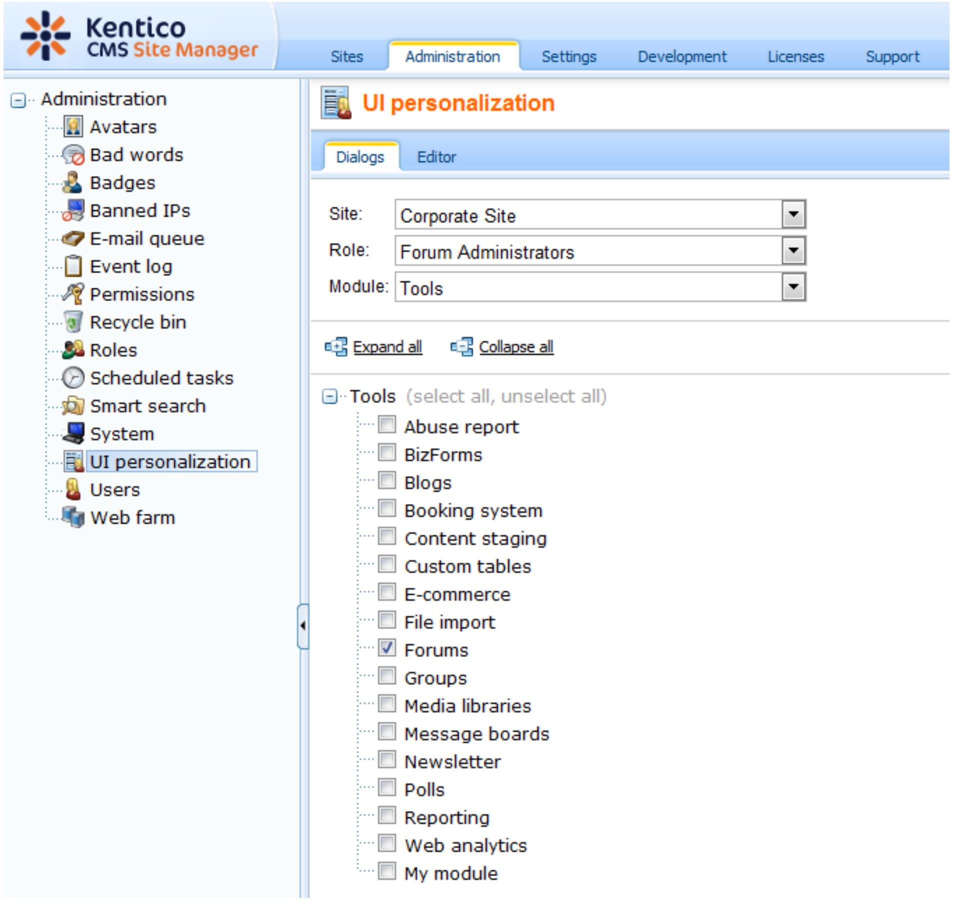


8. On the My desk tab, we will need only the My profile, My messages and My friends sections, all others are not needed for forum administrators. Choose **Module**: My Desk and enable only the UI elements listed below, do not enable the rest.

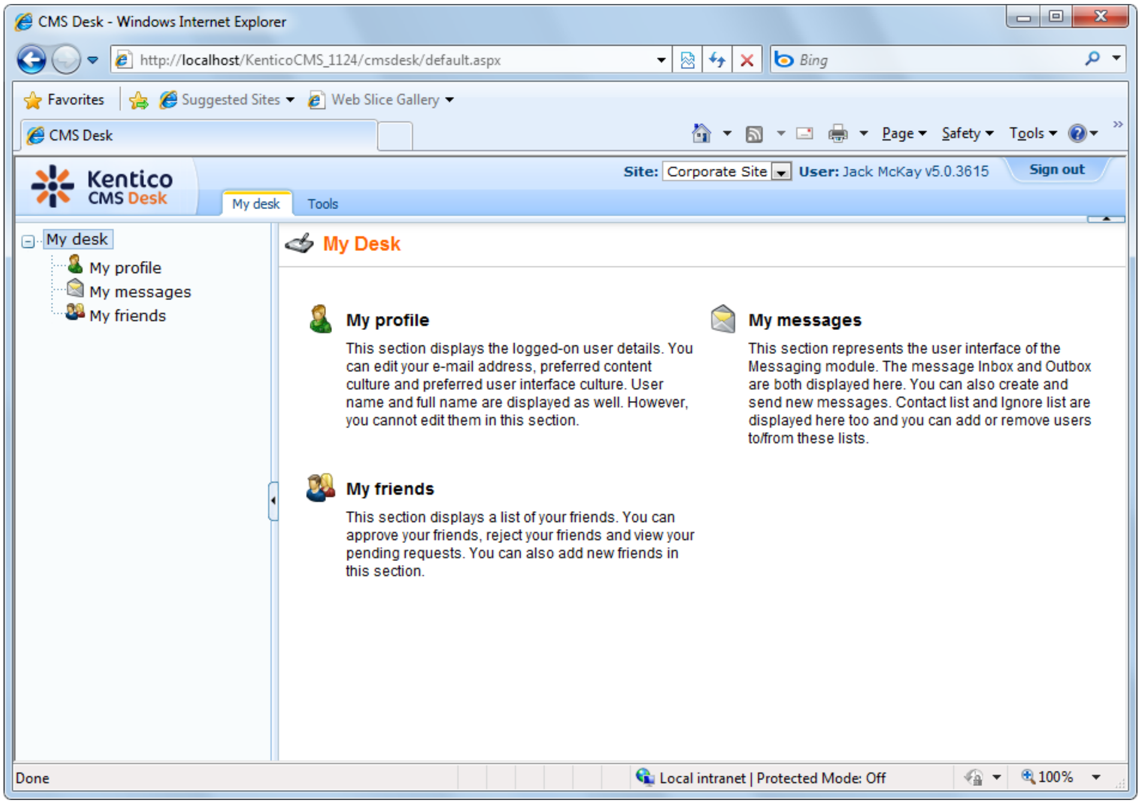
- **My profile:** enabled
  - **Details**: enabled
  - **Change password**: enabled
  - **Notifications**: enabled
  - **Subscriptions**: enabled
- **My messages**: enabled
- **My friends**: enabled



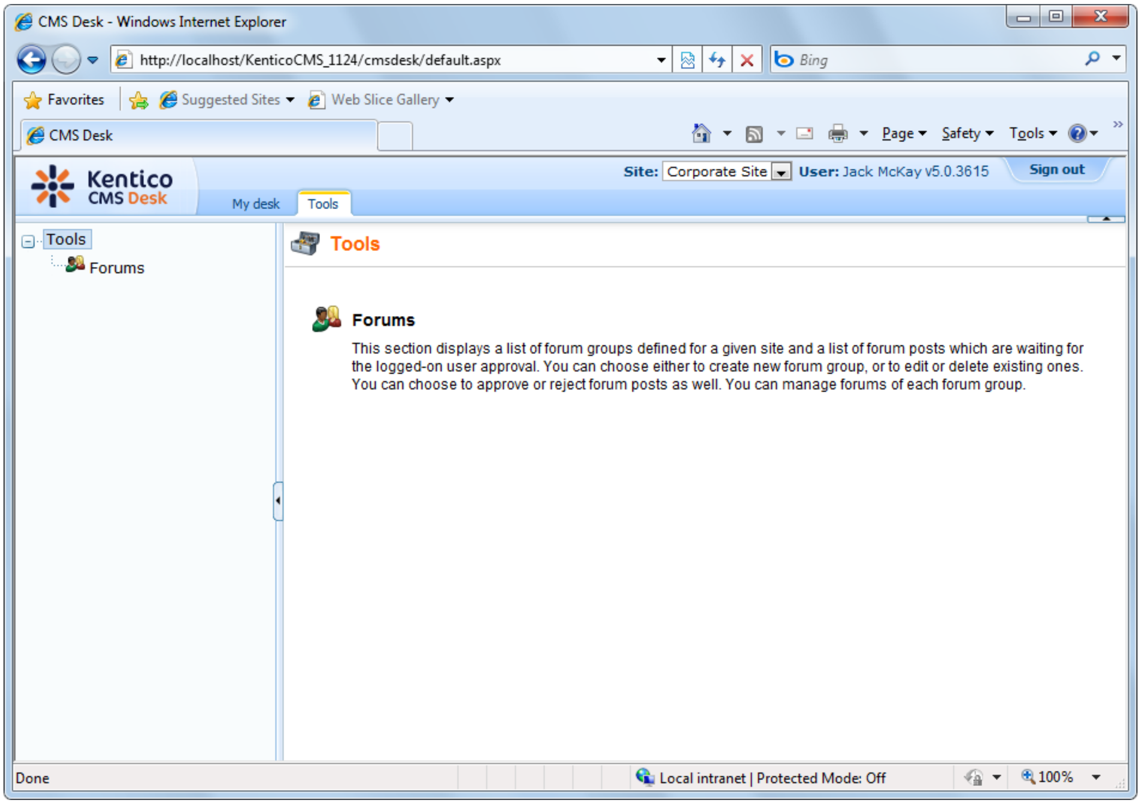
9. Finally, we need only the Forums option in the Tools menu. Choose **Module**: Tools and enable only the **Tools** UI element as you can see in the screenshot below. Leave all the remaining UI elements disabled.



10. We can now verify what we have achieved. Log out of **Site Manager** and log back in to **CMS Desk** as the *ForumsGuy* user created in step 3. You will see the a simplified user interface as in the following screenshot.



If you switch to the **Tools** tab, there is only the **Forums** menu item present. It is evident that this UI is much easier to understand for a first-time end user.



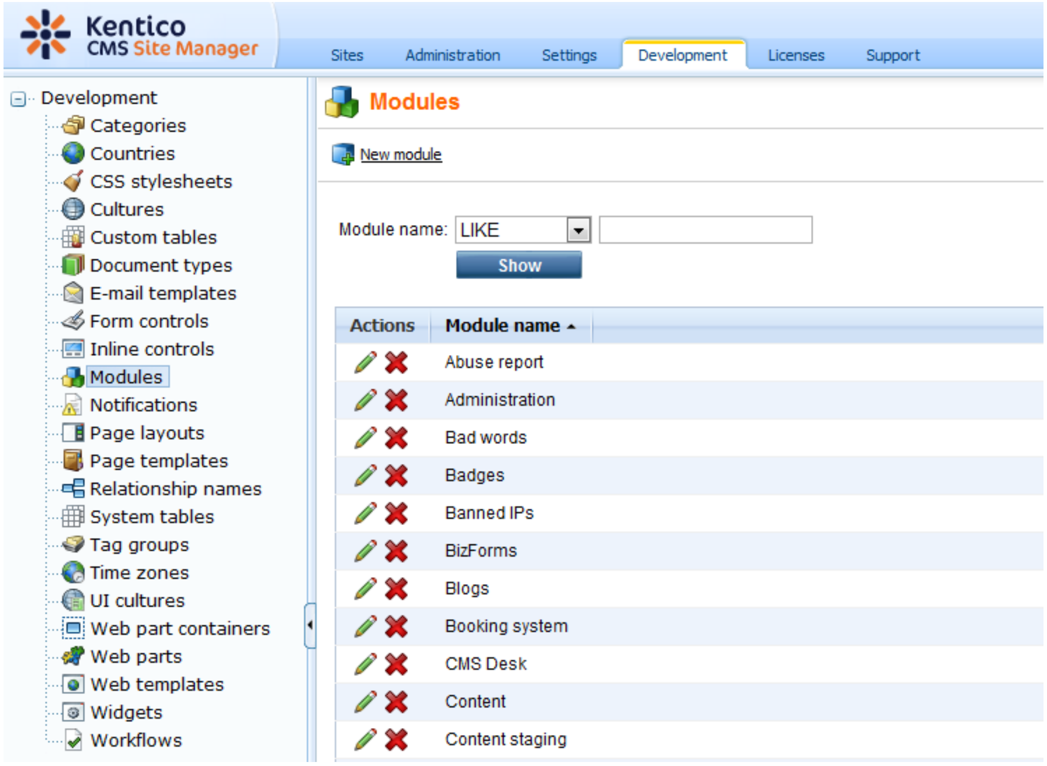
7.10.7.7 UI elements management

7.10.7.7.1 UI elements management overview

The user interface of Kentico CMS consists of modules. Modules contain particular UI elements. UI element is a page or part of a page in CMS Desk which can be hidden from a specified user. It can be:

- **tab**
- **menu item**
- **group of controls**

UI elements of a particular modules can be edited or added in **Site Manager -> Development -> Modules**. This is where you can see a list of all modules in the system. For information on which module represents which parts of the CMS, please read Personalizable parts of CMS Desk.



If you want to manage UI elements of some module, click the **Edit** () icon next to the module and switch to the **User interface** tab. On the **User interface** tab, you can see all UI elements of the module in the tree on the left.

- You can re-order UI elements using the **Up** () and **Down** () buttons. This order will then be reflected in the real UI for tabs and menu items. Parts of pages (groups of controls) can not be re-ordered this way.
- You can delete existing elements using the **Delete element** () button.
- New elements can be created using the **New element** () button (see the example), while the new element is always created under the currently selected node.

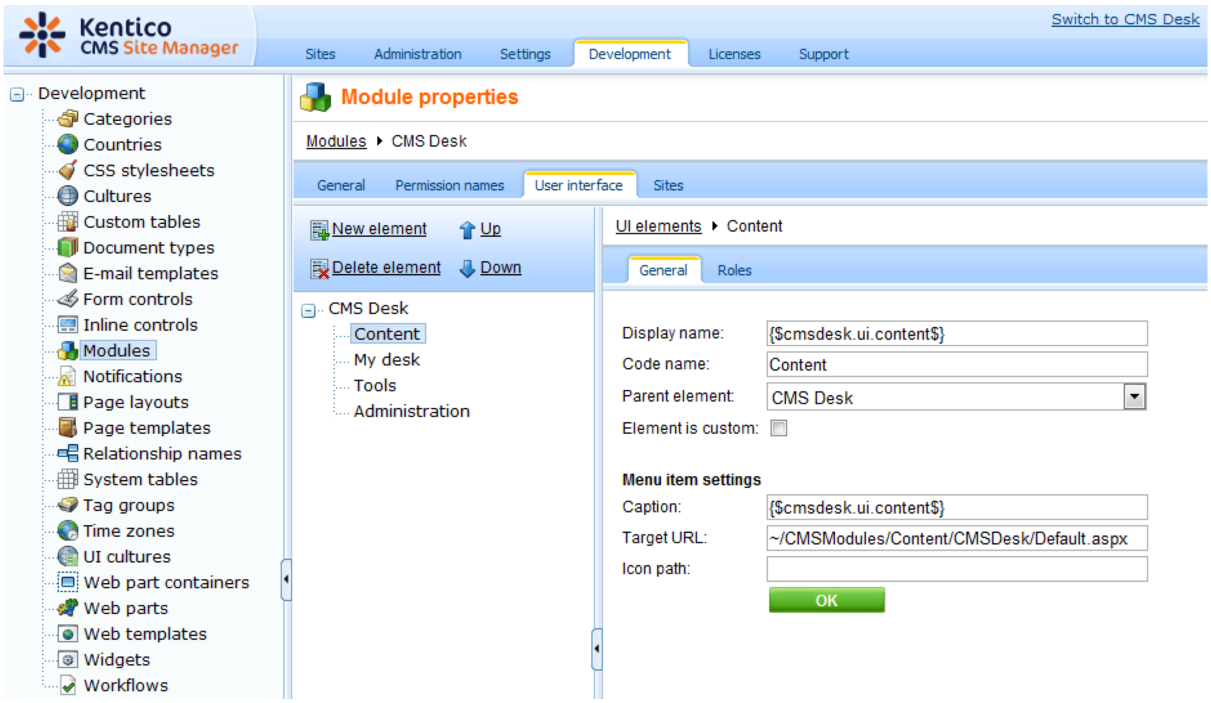
Settings for each UI element are divided into two tabs - **General** and **Roles**.

On the **General** tab, you can set the following properties of the UI element.

- **Display name**: name of the element displayed in the administration interface (i.e. in the settings, not in the final UI); can be entered either as plain text or as a localizable string in the *{$mystringname$}* format
- **Code name**: name of the element used by developers in website code, it must be unique within a module
- **Parent element**: using this drop-down list, you can change the hierarchical position of the element in the UI elements tree for this module; you can select either the name of the module (which places the element under the root) or some of the other UI elements (which places the element under the selected element)
- **Element is custom**: if false, the element is a native part of Kentico CMS; set this value to true for your custom UI elements

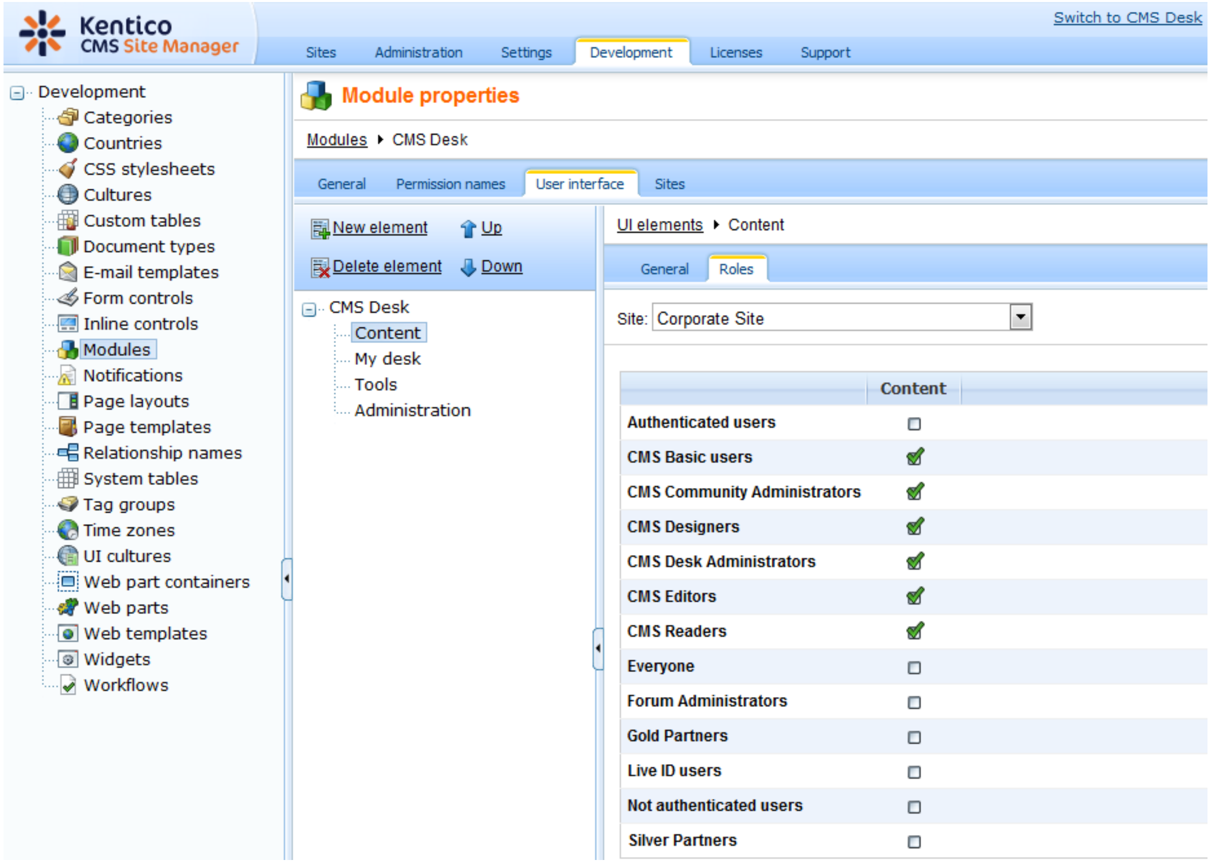
- **Caption**: caption of the UI element displayed in the final UI; can be entered either as plain text or as a localizable string in the *{$mystringname$}* format

- **Target URL**: URL of the page which is the content of the UI element; you can enter both absolute (e.g. http://*www.google.com*) and relative (e.g. *~/CMSModules/Content/CMSDesk/Default.aspx*) URLs
- **Icon path**: icon displayed next to element caption - applicable **only for menu items**; you can enter either a full relative path begining with ~ (e.g. *~/App_Themes/Default/Images/CMSModules/list.png*) or a short path begining under the *Images* folder of the current skin (e.g. *CMSModules/list.png*)

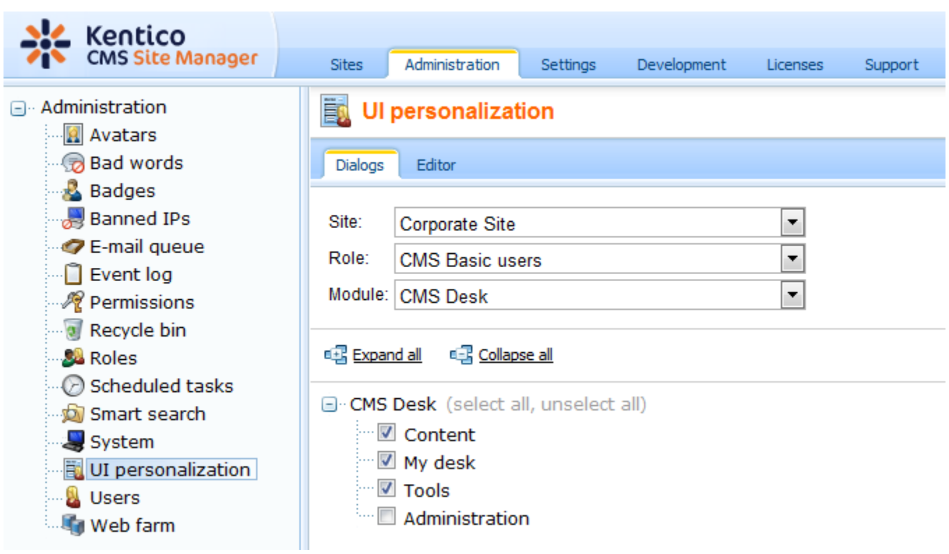


If you switch to the **Roles** tab, you can directly configure access to the selected UI element for particular roles.

- If you enable (✓) a check-box, the UI element will be visible to members of the role.
- If you disable (☐) a check-box, members of the role will not see the UI element (unless they are members of some other role which has it allowed).



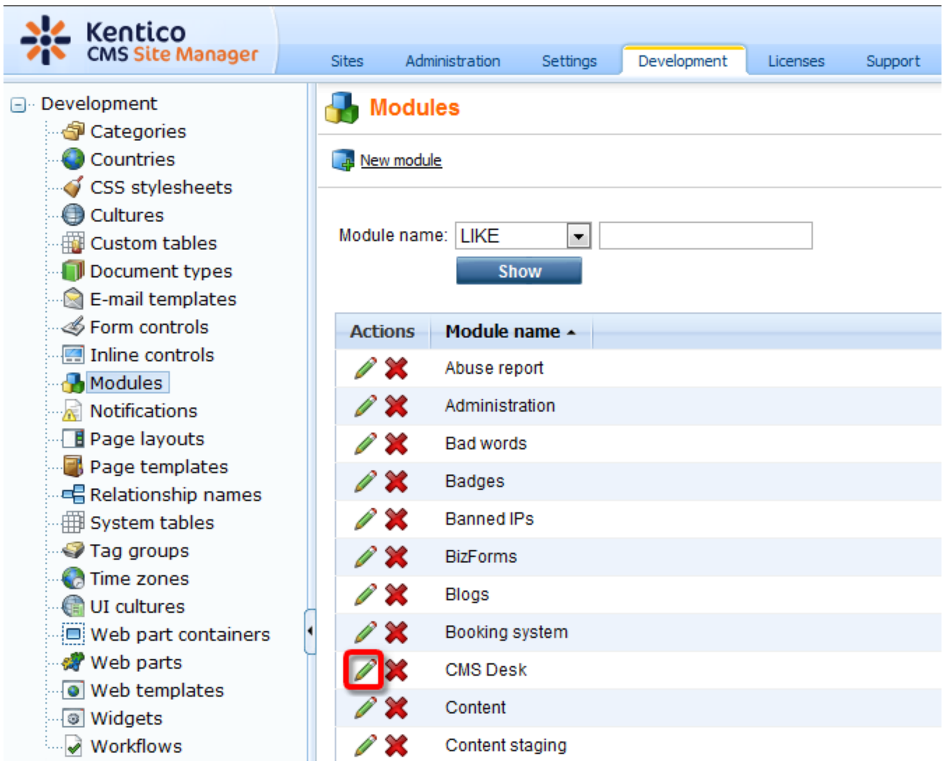
This is just another way how to make the same configuration as described in UI profile configuration, i.e. settings made here are reflected in **Site Manager -> Administration -> UI personalization** (see the screenshot below) and vice-versa.



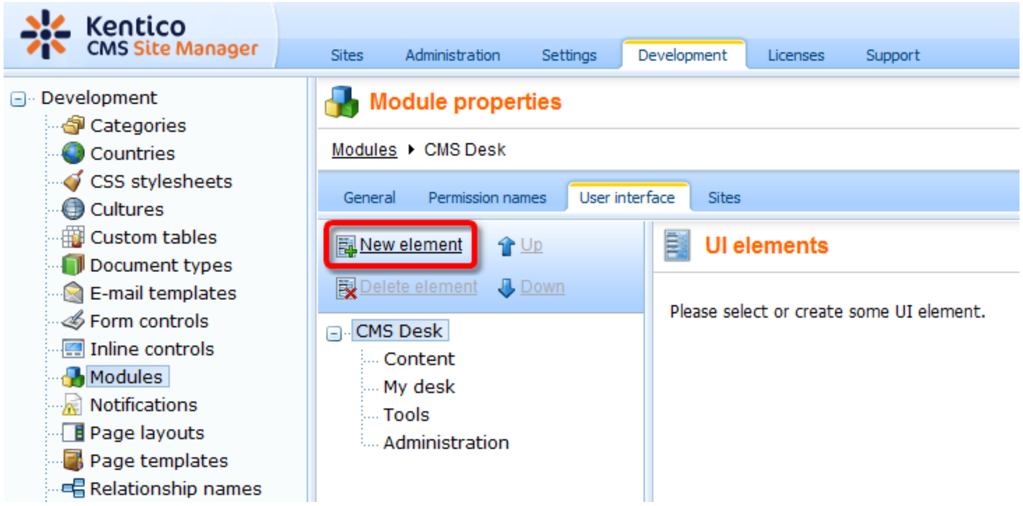
7.10.7.7.2 Example: Adding a new main tab to CMS Desk

In this example, you will see how to add another tab next to the **Content**, **My desk**, **Tools** and **Administration** tabs in **CMS Desk**. This procedure can be used to add your custom UI elements to any other personalizable part of the CMS. You can also integrate your custom modules into the UI this way as described here.

1. We know from this chapter that the main tabs belong to the **CMS Desk** module. Go to **Site Manager -> Development -> Modules** and **Edit** ( ) the **CMS Desk** module.



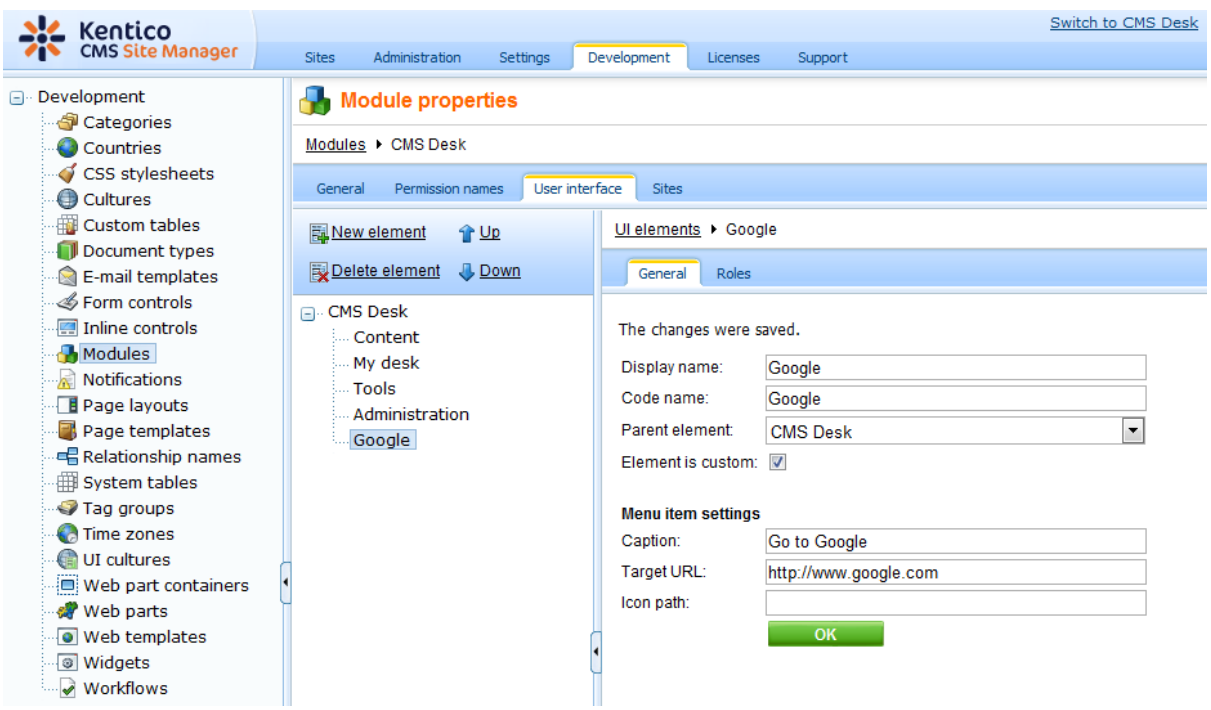
2. In the module's administration interface, switch to the **User interface** tab. You can see the four UI elements representing the tabs in the tree on the left. Select the root of the tree and click the **New element** ( ) button.



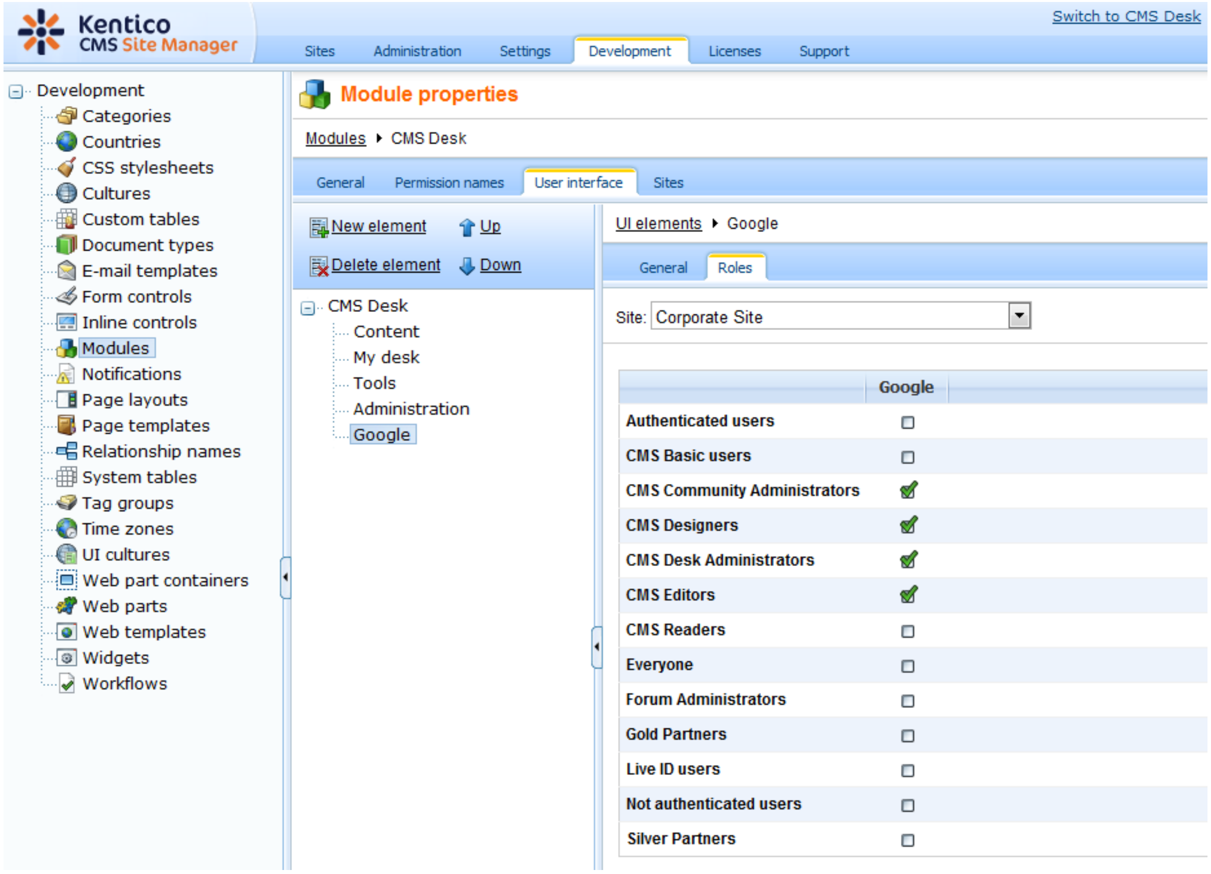
3. In the **New element** dialog, enter the following details:

- **Display name**: Google
- **Code name**: Google
- **Element is custom**: true
  - **Caption**: Go to Google
  - **Target URL**: http://www.google.com
  - **Icon path**: leave blank (icons can be used only for left menu items, not for tabs)

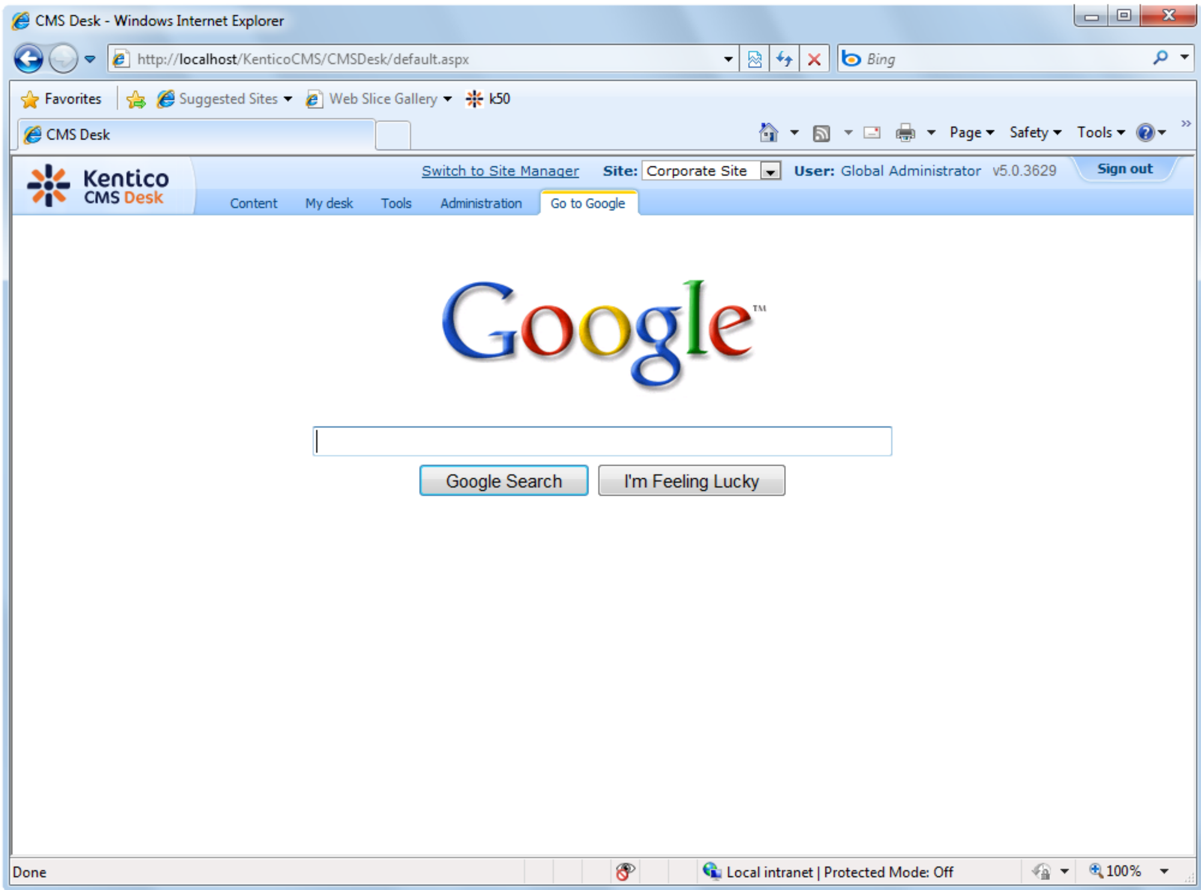
and click **OK**. The new UI element will be added to the tree as in the screenshot below.



4. Switch to the **Roles** tab and enable the UI element for the desired roles.



5. Now switch to **CMS Desk** logged in as a member of one of the roles enabled in the previous step. You will see our new **Go to Google** tab present next to the original four tabs as you can see below. By clicking it, you get the Google title page displayed within the CMS UI.



Please note

If you try to display a page with scroll bars within the CMS UI, you may not be able to view the whole page. This is because the current version of Kentico CMS does not allow **page scrolling** here due to the UI design.
To enable page scrolling within the CMS UI, it is recommended to create a new ***.aspx* page** containing a **frameset with just one scrolling-enabled frame** pointing to your page.

7.10.8 User registration

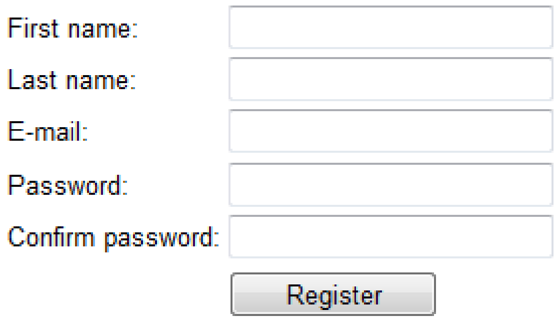
7.10.8.1 Available registration web parts

There are three different ways how you can let site visitors register to your site:

- Using the **Registration form** web part. Read more here.
- Using the **Custom registration form** web part. Read more here.
- Via **Windows Live ID**. Learn more here.

7.10.8.2 Registration form web part

The Registration form web part is a ready-made web part that can be used right out of the box. You can just place it on any page of your website without setting any web part properties. However, if you want to modify the default behavior of the web part, you can set a number of web part properties. You can find a detailed description of these properties in **Kentico CMS WebParts** reference guide.

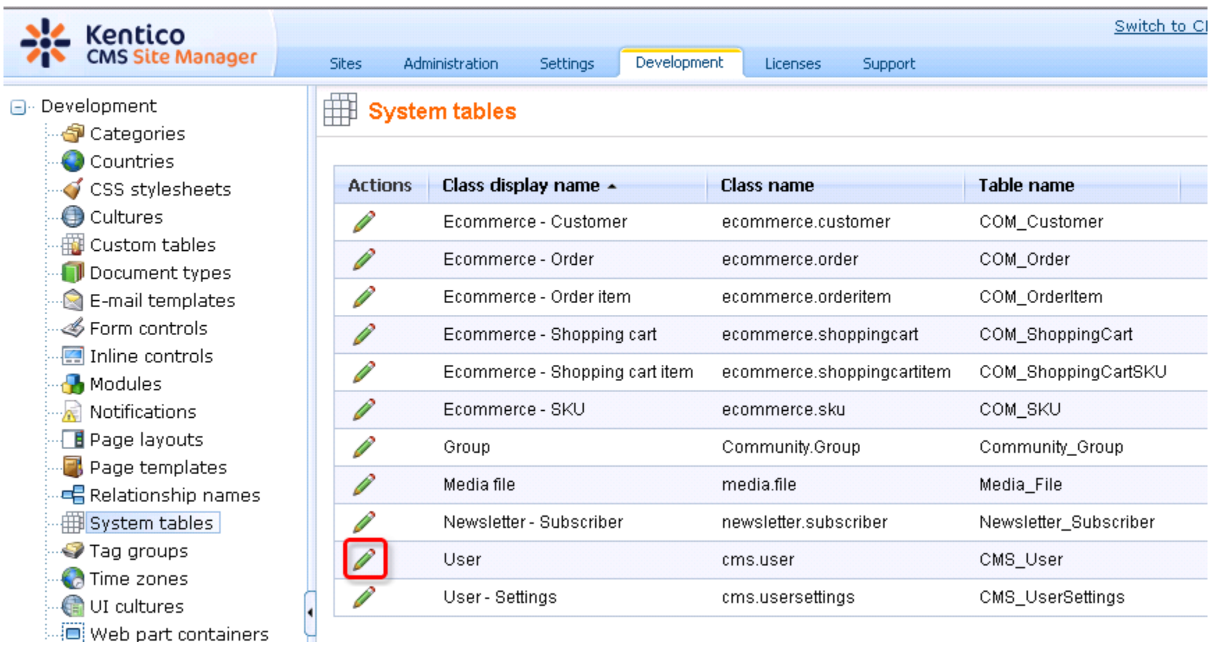


7.10.8.3 Creating a custom registration form

The **Custom registration form** web part can be used in situations when you want to use a different registration form than the default one provided by the **Registration form** web part. This is typically when you want users to provide different details or when you want to customize the form's layout.

In the following example, you will learn how to use a custom registration form on your site. You will create an **alternative form** and use it for registration via the **Custom registration form** web part. If you are not familiar with the **Alternative forms** concept, please refer to the Alternative forms chapter first.

1. Go to **Site Manager -> Development -> System tables** and choose to **Edit** ( ) the **User** system table.

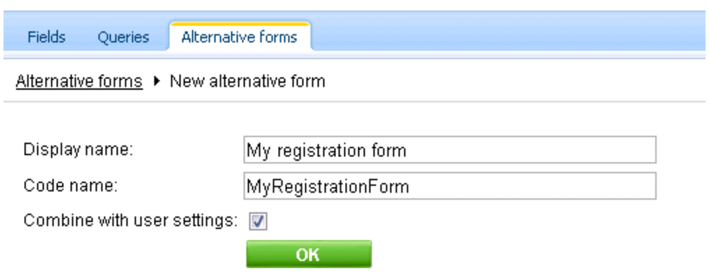


2. Switch to the **Alternative forms** tab and click the **Create new form** link above the list.

3. Fill in the following details:

- **Display name -** My registration form
- **Code name -** MyRegistrationForm
- **Combine with user settings -** make checked; this ensures that all user's fields will be available

and click **OK**.

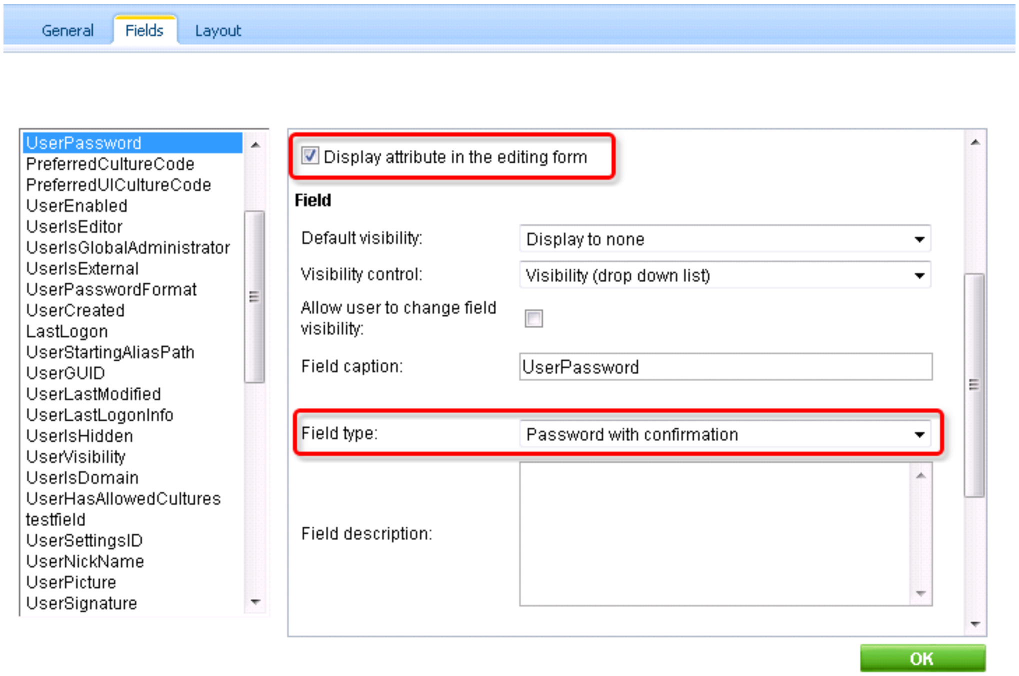


4. Your new form is now created and you are redirected to the editing interface of it. Switch to the **Fields** tab.

On this tab, you can see the attributes defined in the **User** system table. You can select an attribute from the list on the left. In the right part, you can modify its properties. We will want our registration form to contain the following fields:

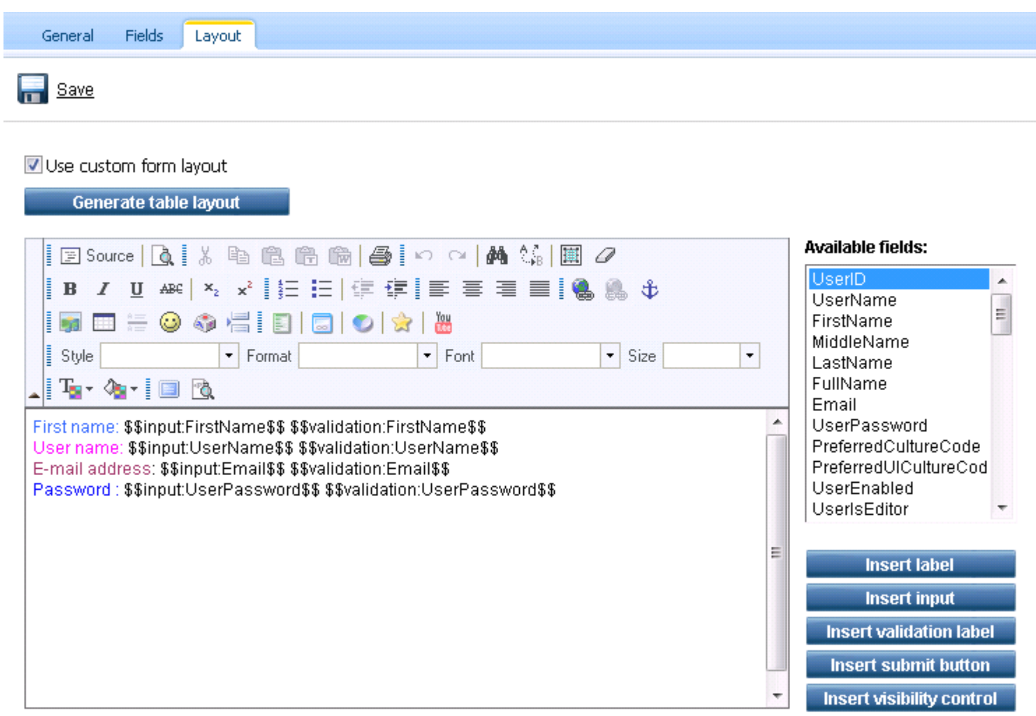
- **UserName**
- **FirstName**
- **Email**
- **UserPassword**

Go through the attributes and check the **Display attribute in the editing form** check-box for those mentioned above, which makes them visible on our registration form. Uncheck the check-box for the rest. For the **UserPassword** attribute, also change the value of the **Field type** drop-down list to **Password with confirmation**.



5. You can also switch to the **Layout** tab and define the registration form's layout using the built-in WYSIWYG editor.

To do it, check the **Use custom layout** check box and click the **Generate table layout** button. A default layout will appear in the editor. Try playing around a bit with the layout or just create something similar to what you see in the screenshot below. Click **Save** when you are finished.



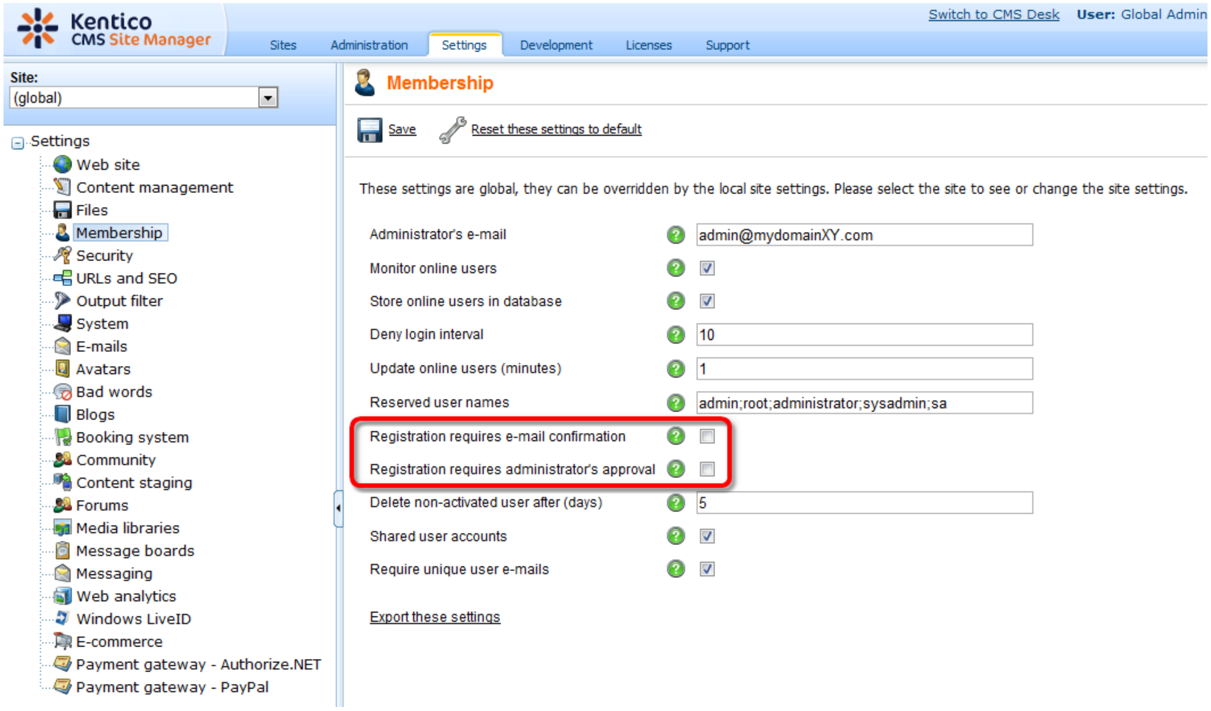
6. Now switch to **CMS Desk**. Choose or create a page where you want to add the registration form and select it in the content tree. Switch to the **Design** tab and add ( ) the **Membership -> Custom registration form** web part to the page. Set the following property of the web part:

- **Alternative form** - cms.user.MyRegistrationForm

7. If you switch to the live site now and go to your page with the Custom registration form web part, you should see the custom registration form that you created. You can now try to register to your website using the form and verify that the user has been created in **Site Manager -> Administration -> Users**.

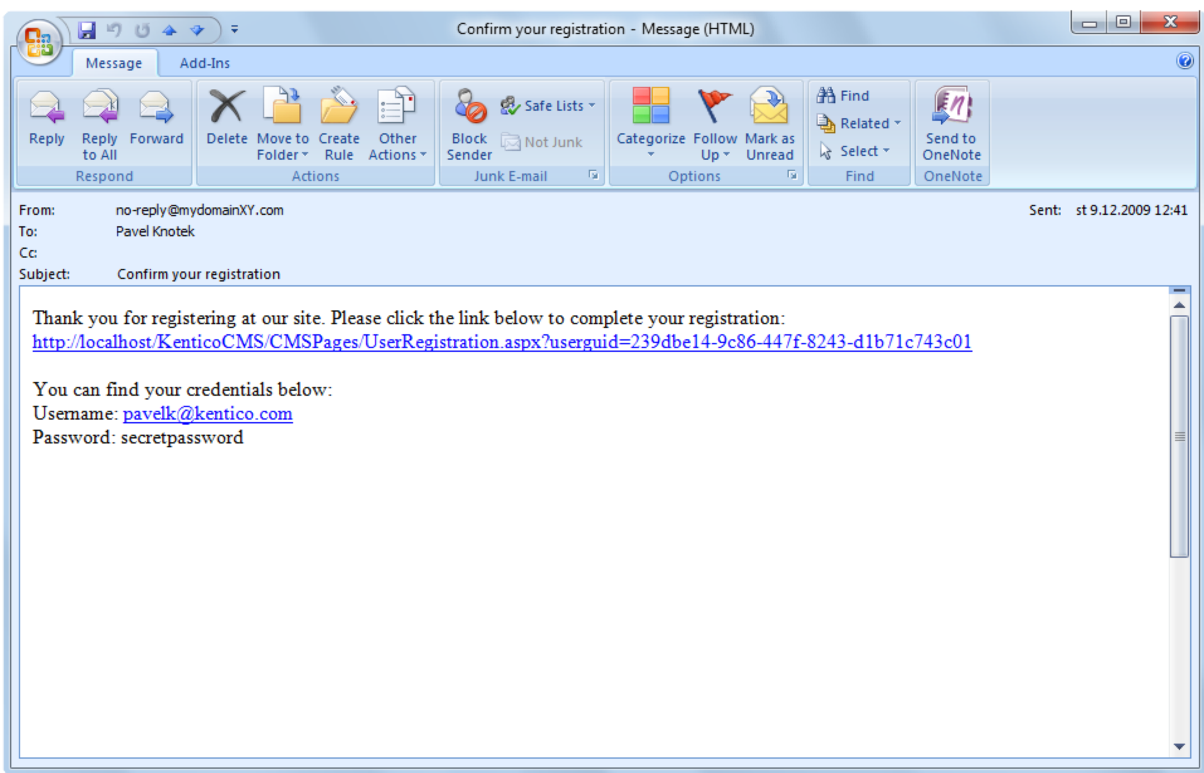
7.10.8.4 Registration approval and double opt-in

By default, users can sign-in to the site immediately after successful registration. However, the two options highlighted in the following screenshot can be enabled in **Site Manager -> Settings -> Membership**. By enabling these options, you can include additional steps in the registration procedure.



Registration requires e-mail confirmation

If checked, newly registered users will receive a confirmation e-mail to the e-mail address specified during registration. This e-mail contains a confirmation link that has to be clicked in order to activate the account. The e-mail is based on the **Membership - Registration confirmation** e-mail template.



After clicking the link, the user will be redirected to the page specified in the **E-mail confirmation page** property of the web part used for registration. This page must contain the **Registration e-mail confirmation** web part to work correctly.

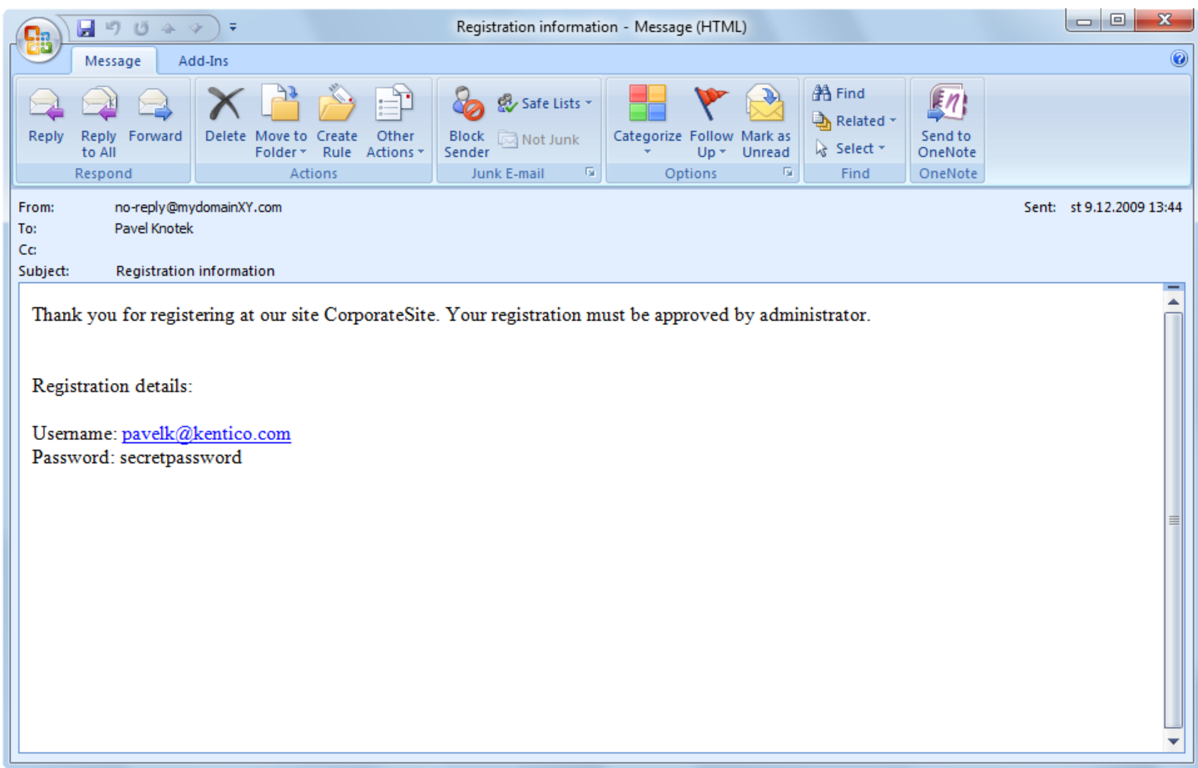
By default, the *~/CMSPages/Dialogs/UserRegistration.aspx* page is used, which displays the following message:

> Your user account is now active. You can sign in using your user name and password. [Click here to continue].

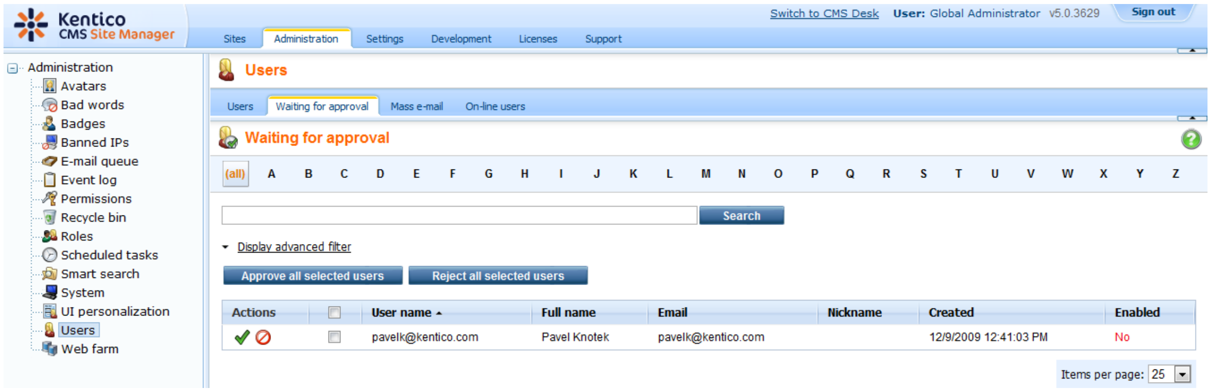
The link at the end of the message will redirect the user to the title page of the website. The user can then log in using the registration details received in the e-mail.

Registration requires administrator's approval

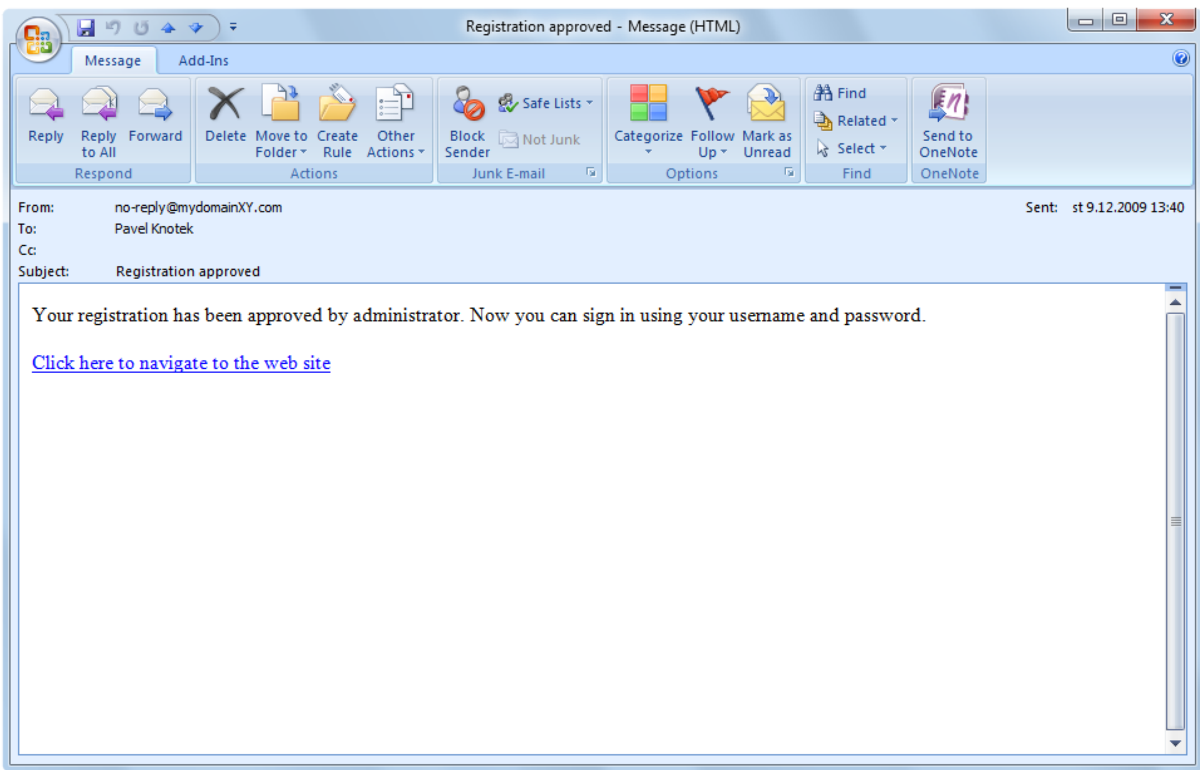
If this option is enabled, users will not be able to sign-in immediately after registration. Their registration will have to be approved by the site administrator. At this point, users will receive an e-mail based on the **Membership - Registration waiting for approval** e-mail template. You can see the default version of the e-mail in the screenshot below.



If the option is enabled, the **Waiting for approval** tab will be displayed in **Site Manager -> Administration -> Users**. On this tab, site administrators can **Approve** ( ) or **Reject** ( ) a user's registration.



After the administrator's approval, users receive another e-mail, confirming that their account has been approved and can be used. The e-mail is based on the **Membership - Registration approved** e-mail template. You can see the default appearance of the e-mail in the screenshot below.



Enabling both options

In case that you enable both of the options mentioned above, the e-mail with the confirmation link will be sent first. After the user's confirmation, registration will have to be approved by the administrator.

> **Please note**
>
> If you have one or both of the options enabled, it is important to properly set the

Redirect to URL property of the web part used for registration. This means that users should not be redirected to any page displaying information about their user account (like the **Members -> Profile** page on the sample Community Starter site), because the account is not active yet (it is waiting for e-mail activation or approval). Such a page would display an error message, which might be misleading for the users.

Third-party authentication issues

Using double opt-in or registration approval in combination with Third-party authentication services may cause certain problems for first-time users. In these cases, new users are by default created without an e-mail address when they log in for the first time. This means that they cannot activate their account via e-mail confirmation or receive notifications informing that their account must be approved by an administrator.

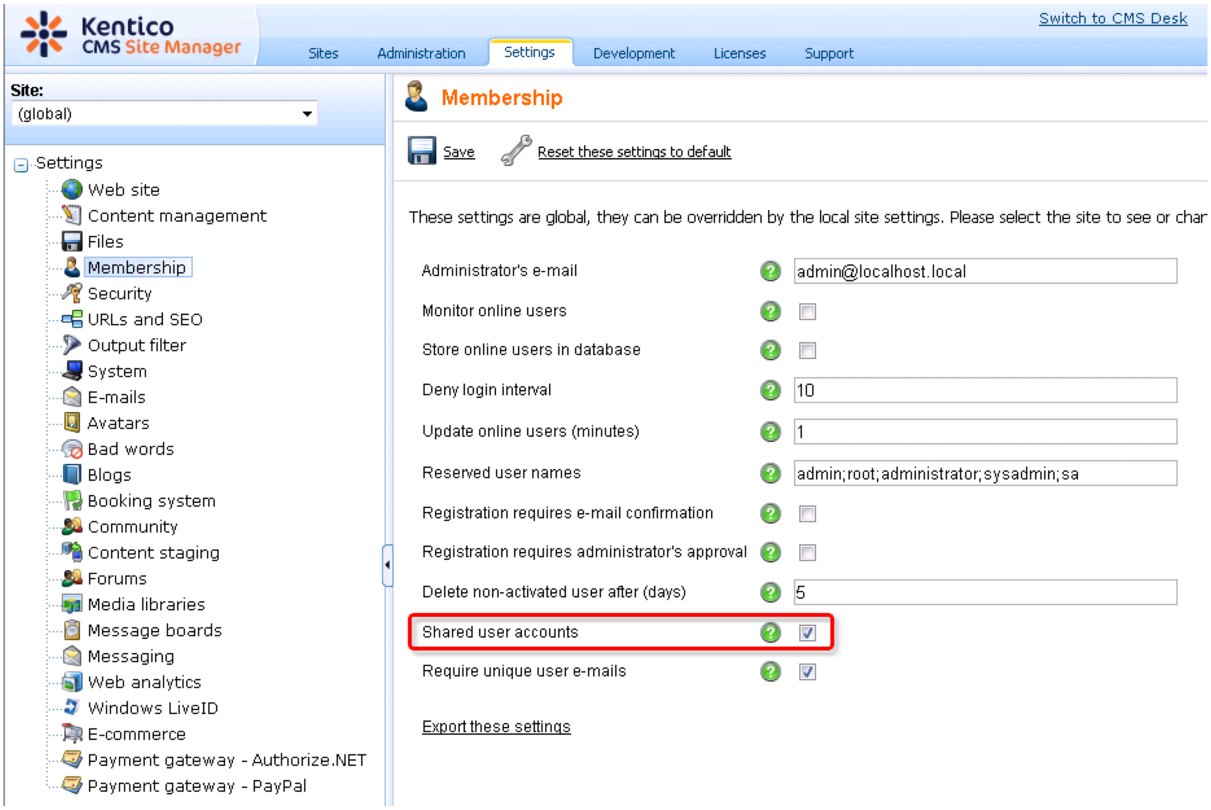
These issues can be avoided by creating a **Required user data page** where users must enter an e-mail address for their account, as described in the chapter linked above.

7.10.8.5 Shared user accounts

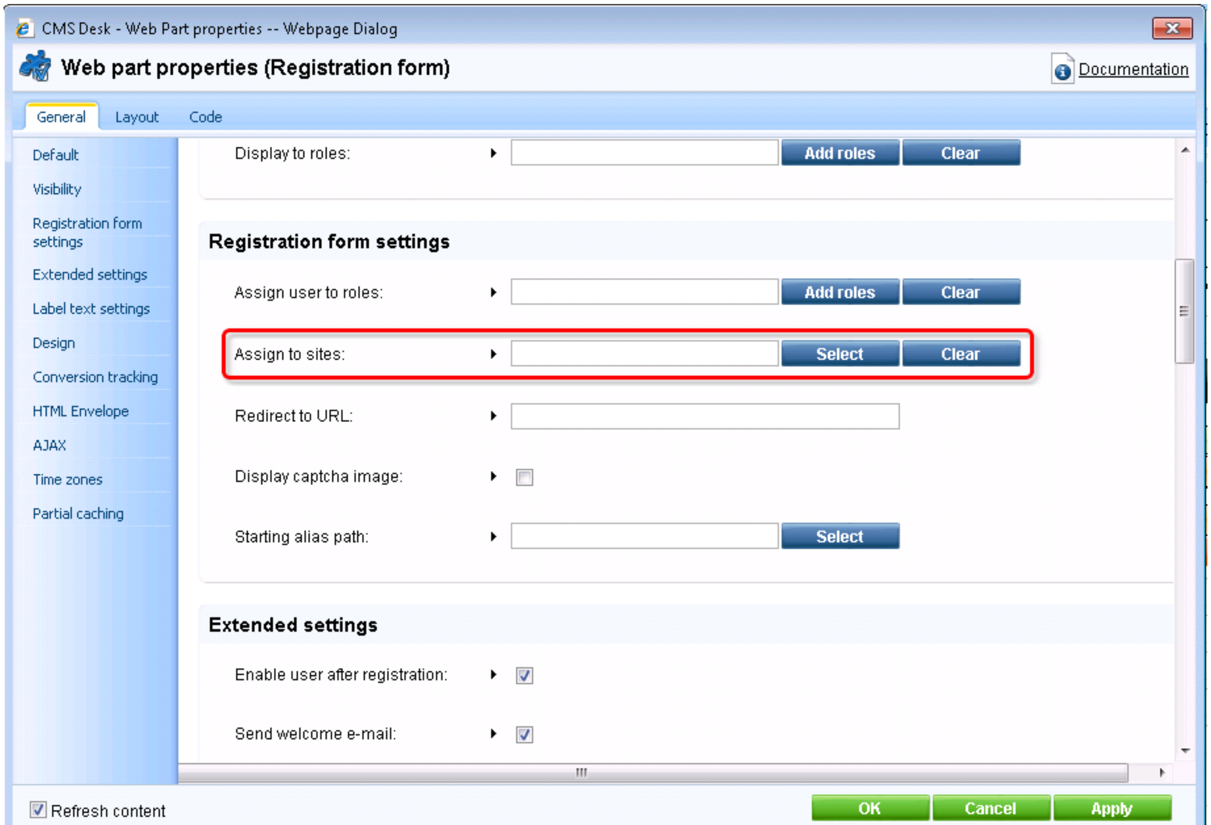
User accounts can be shared among all sites running on one Kentico CMS installation. This means that if a user creates an account on one site, they can automatically log-on to the other sites running on the same installation using the same credentials.

This behaviour can be switched on or off in **Site Manager -> Settings -> Membership**, using the **Shared user accounts** check-box.

- If the check-box is enabled, user accounts created on one site will be shared among all the sites running on the installation.
- If the check-box is disabled, new accounts will be assigned only to the current site and not the others.



However, registration web parts have the **Assign to sites** property. Using this property, you may determine which sites the user accounts created via the web part will be assigned to.



7.10.9 Badges

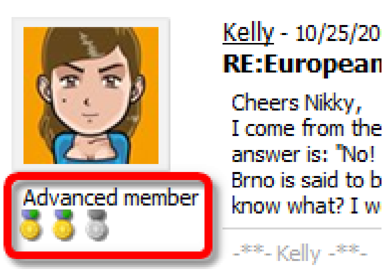
7.10.9.1 Badges

Users can be labeled with badges. These are images with a short text, expressing a user's activity level, importance or role in the context of the website. They are displayed in forum posts, on user public profiles or in your own custom controls.

There are two types of badges:

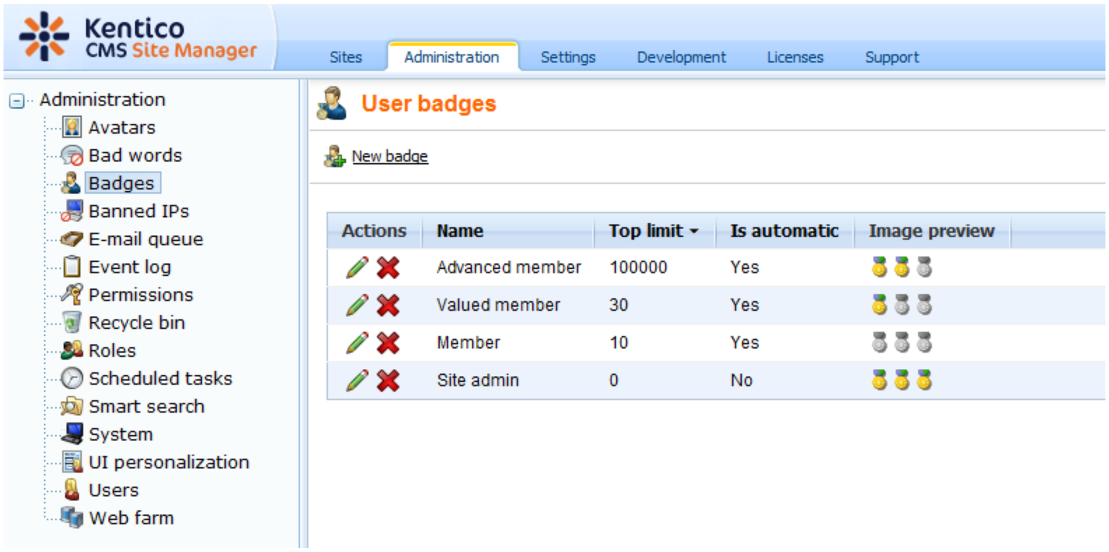
- **Automatic badges** - these are assigned to users based on the number of gained activity points
- **Non-automatic badges** - these are assigned to users manually by site administrators and are assigned permanently, regardless of the number of gained activity points

In the screenshot below, you can see one of the pre-defined badges in a forum post.



7.10.9.2 Defining badges

Badges can be defined in **Site Manager -> Administration -> Badges**. On this page, you can see a list of currently defined badges. You can **Edit** ( ) or **Delete** ( ) badges in the list or define a new badge by clicking the **New badge** link at the top part of the page.



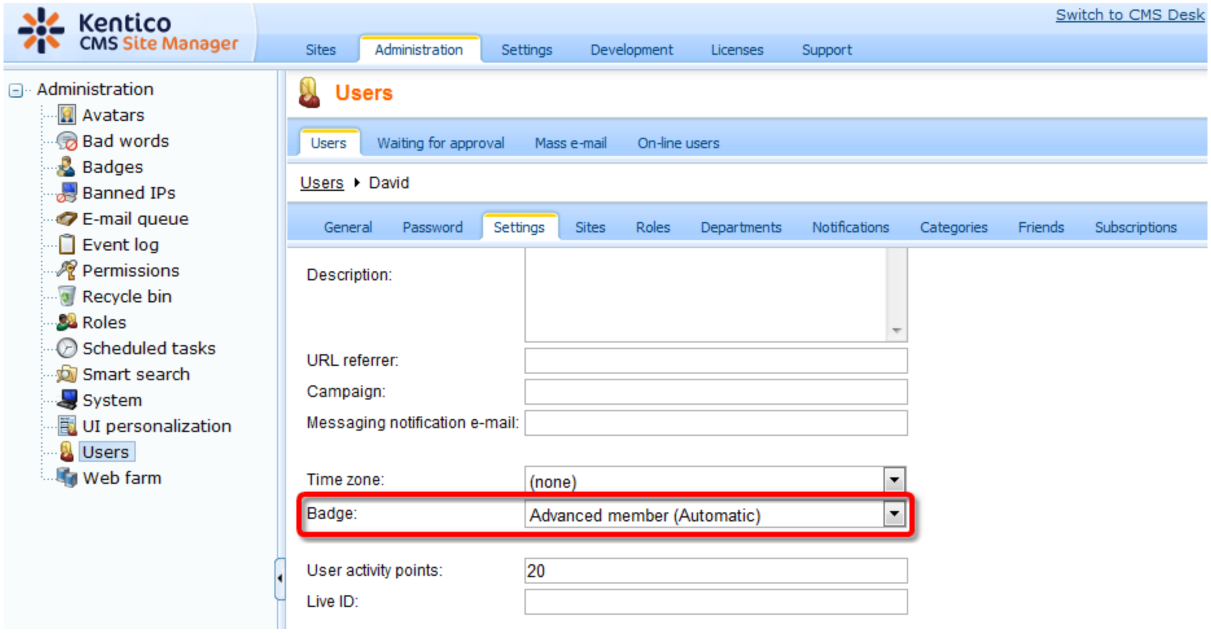
When creating a new badge, the following properties can be defined:

- **Display name** - name of the badge displayed in the administration interface and on the site
- **Code name -** name of the badge used in code
- **Image URL** - URL of the badge's image
- **Is automatic** - if checked, the badge will be assigned to users automatically based on the number of gained activity points; if unchecked, the badge can be assigned to users by site administrator and will remain assigned permanently, regardless of the number of gained activity points
- **Top limit -** number of activity points required for the user to get the badge; applies only for automatic badges

7.10.9.3 Assigning badges to users

Site administrators can assign badges to users in **Site Manager -> Administration -> Users -> Edit ( ) user -> Settings**. It can be done by the **Badge** drop-down list.

This is typically used to assign users with non-automatic badges. However, automatic badges can be assigned to users this way too.

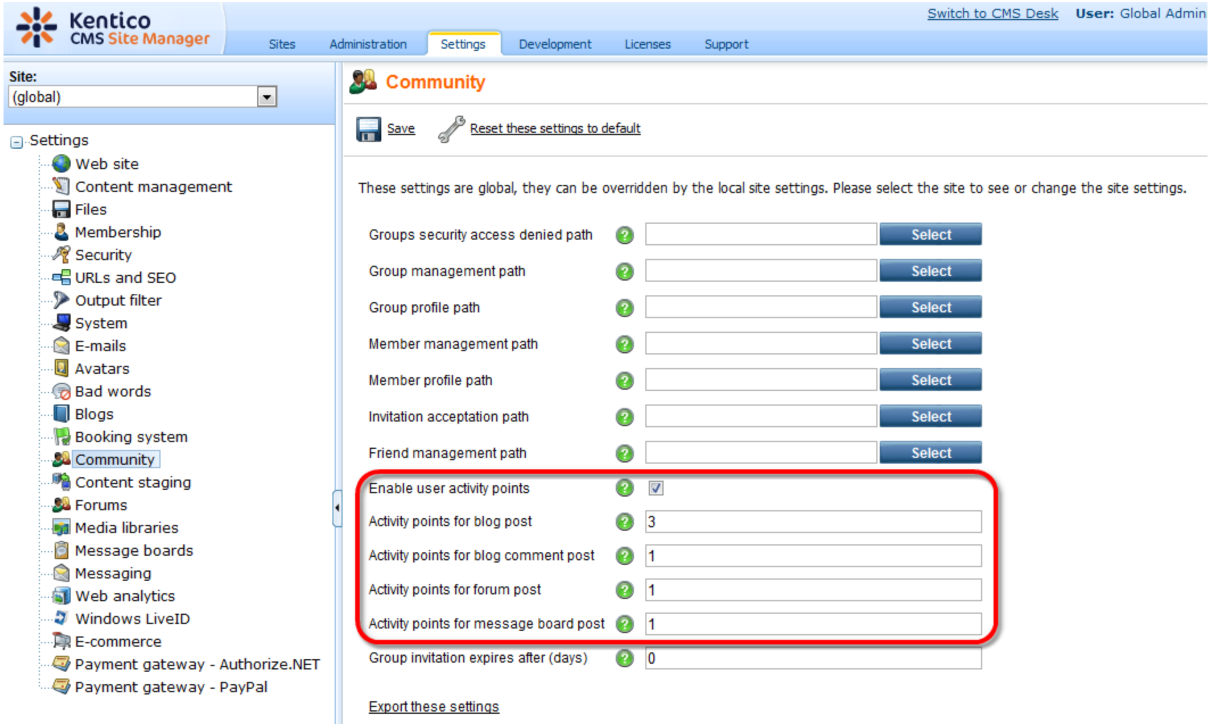


7.10.9.4 Activity points

Users can gain activity points for their activity on the site. Based on the number of gained activity points, user can be assigned with automatic badges. For this feature to be functional, you have make the following settings in **Site Manager -> Settings -> Community**:

- **Enable user activity points** - enables the activity points feature

- **Activity points for blog posts** - number of activity points that users receive for adding a blog post
- **Activity points for blog comment post** - number of activity points that users receive for adding a blog post comment
- **Activity points for forum post** - number of activity points that users receive for adding a forum post
- **Activity points for message board post** - number of activity points that users receive for adding a message board post



7.10.9.5 Available form controls

The following form controls can be used in your custom controls to display users' badges:

- **Viewer - Badge image (ViewBadgeImage) -** used for displaying image of a badge
- **Viewer - Badge text (ViewBadgeText) -** used for displaying 'Display name' of the badge

7.10.10 Custom fields visibility

7.10.10.1 How it works

The visibility controls functionality is designed to enable registered users to decide which fields of their public profile will be visible to other users. You can find an example of how this works on the sample **Community site**.

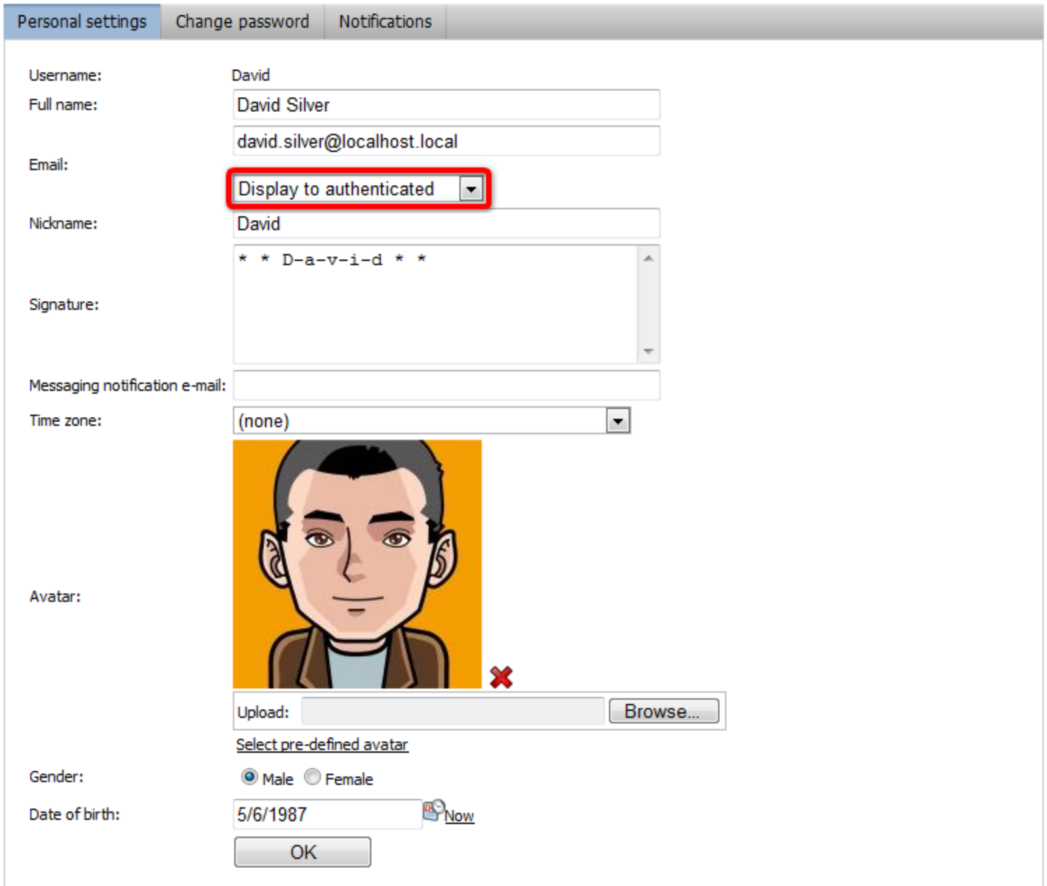
1. Run the **Community site** and sign out of the administration interface. Log on to the site using the **Sign in** form on the right. Enter *David* with blank password and click **Log on**.

2. Once logged in, **Shortcuts** menu will be displayed where the Sign in form previously was. Click the **Edit my profile** link, you will be redirected to the user's profile editing page.

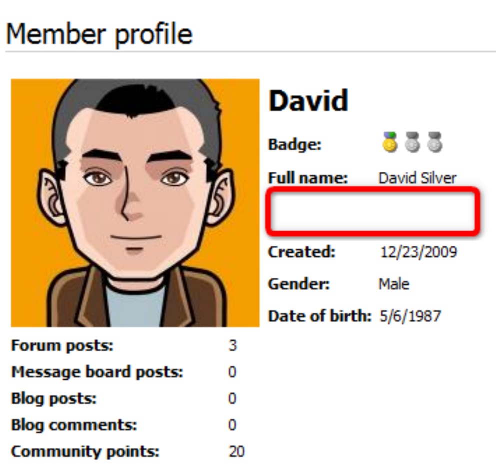
3. You can see a drop down list next to the **E-mail** field, as in the screenshot below. Using this control, users can define to whom will the e-mail address be displayed. The following four options are available:

- **Display to none** - the field will not be displayed to anyone
- **Display to all** - the field will be displayed to everyone
- **Display to authenticated** - the field will be displayed only to authenticated users
- **Display to friends** - the field will be displayed only to the user's friends

Choose **Display to authenticated** and click **OK**.

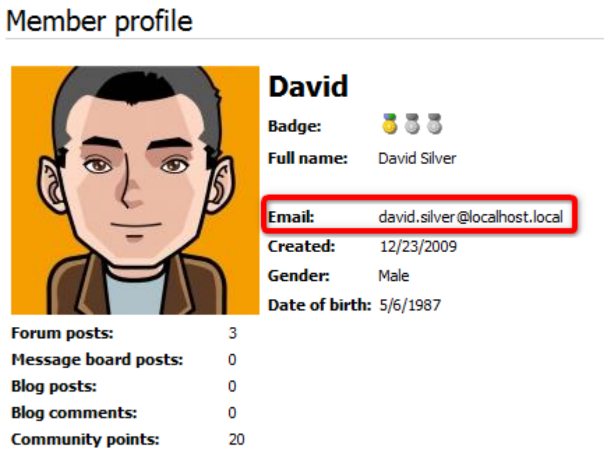


4. You have just configured the user's profile so that only authenticated users can see their e-mail address. Let's verify that it really works. Sign out and visit David's profile as an unauthenticated site visitor. From the site's main menu, select Members and click David's icon in the list below. You should see his profile, but the e-mail address is not present.



5. Now sign in the same way as you did in step one, but use *Mia* with blank password instead. This signs you in as another registered member of the site. And these are the ones who should see the e-

mail address, aren't they? So once signed in, view David's profile again. And ... Bingo!!! ... the e-mail address is visible.



7.10.10.2 Enabling visibility controls

The visibility selection drop-down list can be added to any field of the user's profile, not only the e-mail field as shown in the example in the How it works topic. This can be changed in **Site Manager -> Development -> System tables**. Choose to **Edit** () the **User (CMS_User)** system table and switch to the **Alternative forms** tab.

If you are not familiar with the **Alternative forms** concept, please read the Module Alternative forms chapter first.

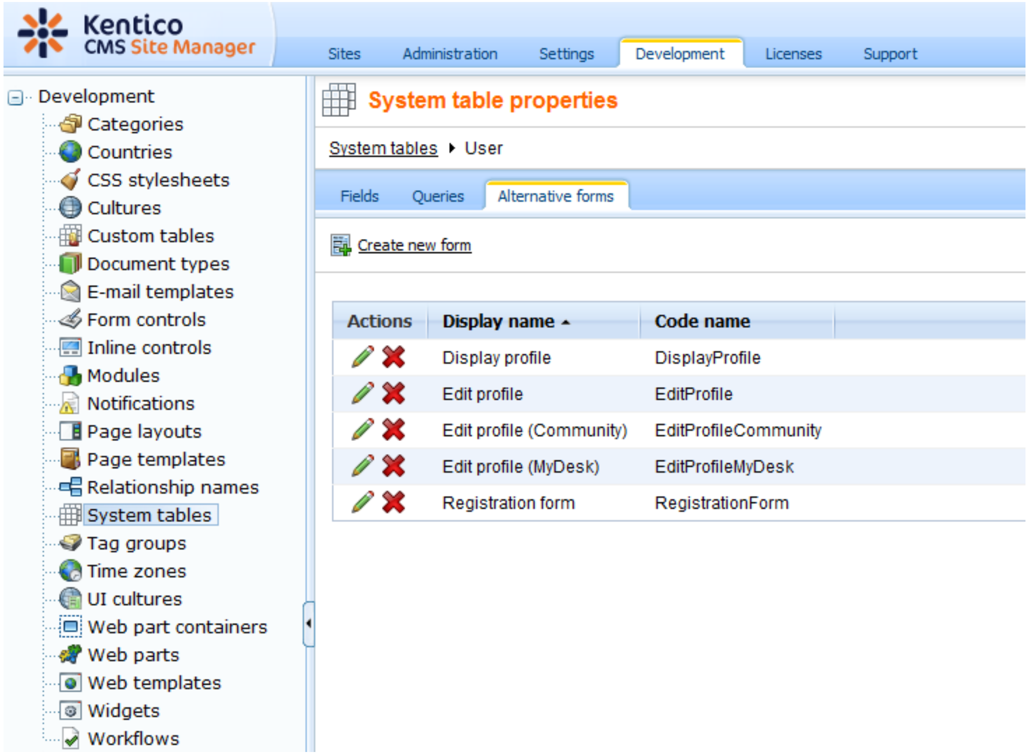
The **CMS_User** system table is the database table where information about registered users is stored. Each of the four alternative forms that you see in the list is used in a specific situation when the system accesses the table:

- **Registration form** - when registering a new user using the **Custom registration form** web part
- **Display profile** - when displaying a user's public profile using the **User public profile** web part
- **Edit profile** - when editing a user's profile using the **My account** web part
- **Edit profile (MyDesk)** - when editing the user profile in **CMS Desk -> My Desk -> My profile**

The **Community_Group** system table is the database table where information about groups is stored. The table has the following alternative form:

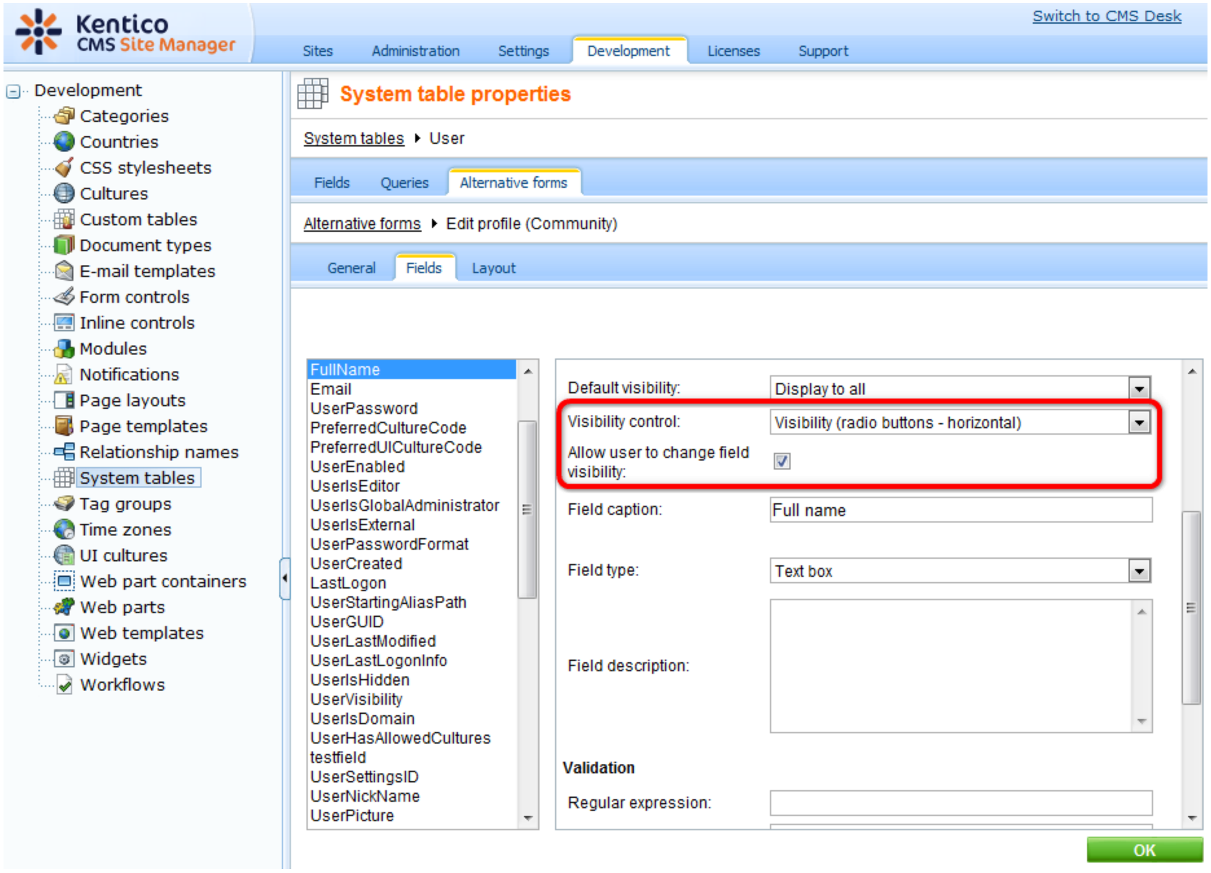
- **Display profile** - displaying group's public profile using **Group public profile** web part

The visibility field can be set in each of these forms.



Let's presume that we want the **Full name** to be optionally hidden in users' public profiles, based on the users' decisions made in their profile editing section.

1. Choose to **Edit** (✎) the **Edit profile (Community)** alternative form, which is the form that is used by the **My profile** web part on the Community site. Switch to the **Fields** tab and select **FullName** from the list on the left. Select **Visibility (radio buttons - horizontal)** from the **Visibility control** drop-down. Check the **Allow user to change field visibility** check-box and click **OK**.

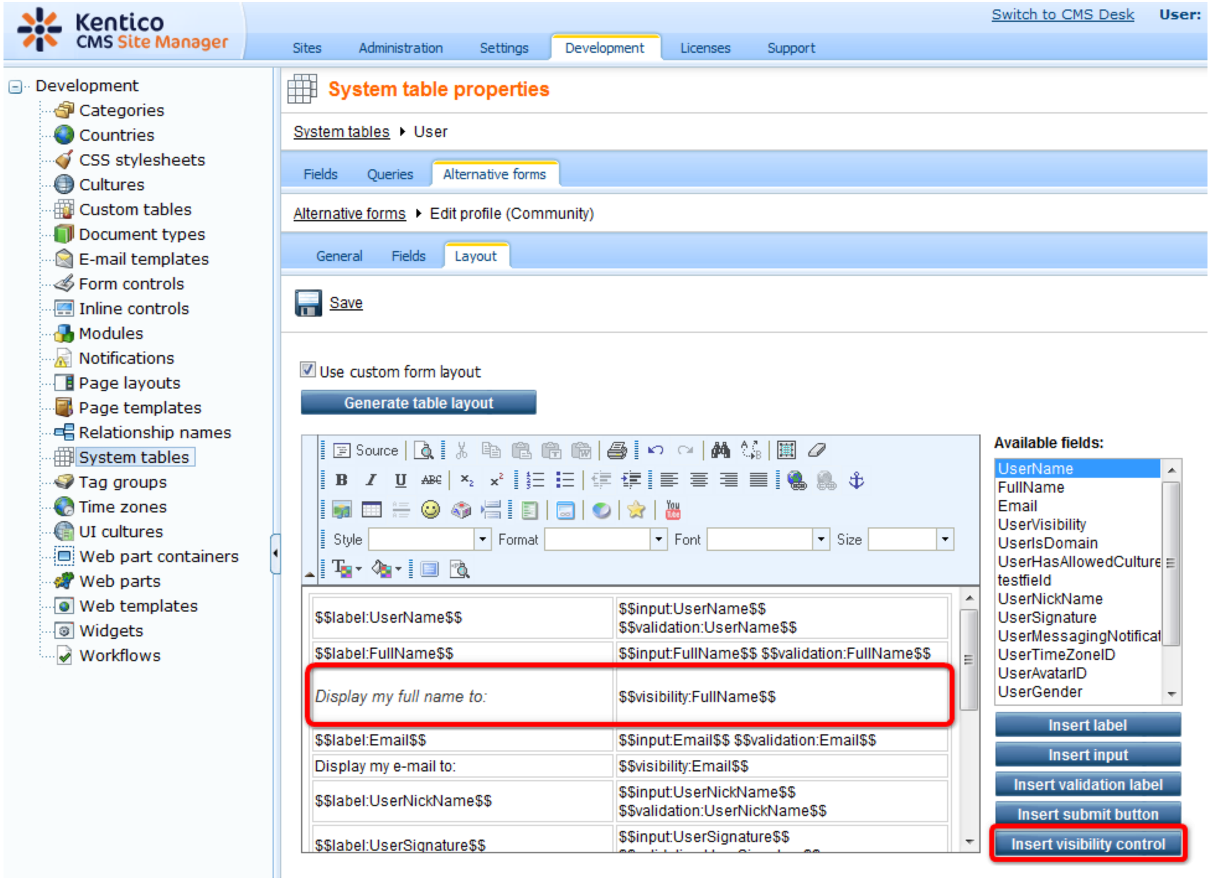


2. Switch to the **Layout** tab. We will need to create another line in the table and add the visibility control into it.

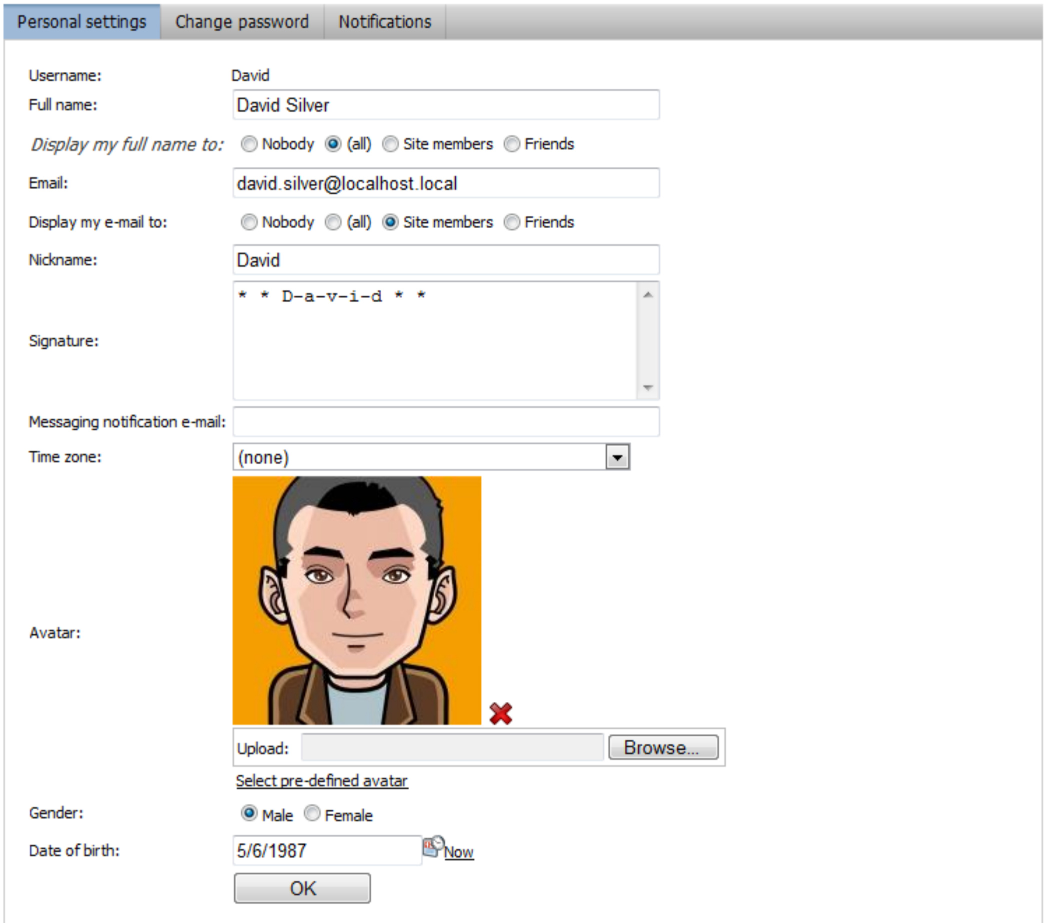
Place the cursor into the FullName line (where the $$label:FullName$$ value is). Right-click and choose **Row -> Insert Row After** from the context menu. Into the first column of the new row, enter *Display my full name to:*.

Now we will add the visibility control itself. Place the cursor into the second column, select **FullName** from the **Available fields** list and click the **Insert visibility control** button. The result should look as in the screenshot below.

When you are finished, click **Save**.



3. Now sign out of the administration interface and sign in to the website as *David* with blank password. Click the **Edit** my profile link in the right **Shortcuts** menu. You should see a drop-down list next to the full name field, so that now the users can determine the visibility of the Full name field.



7.10.10.3 Use in custom form layouts

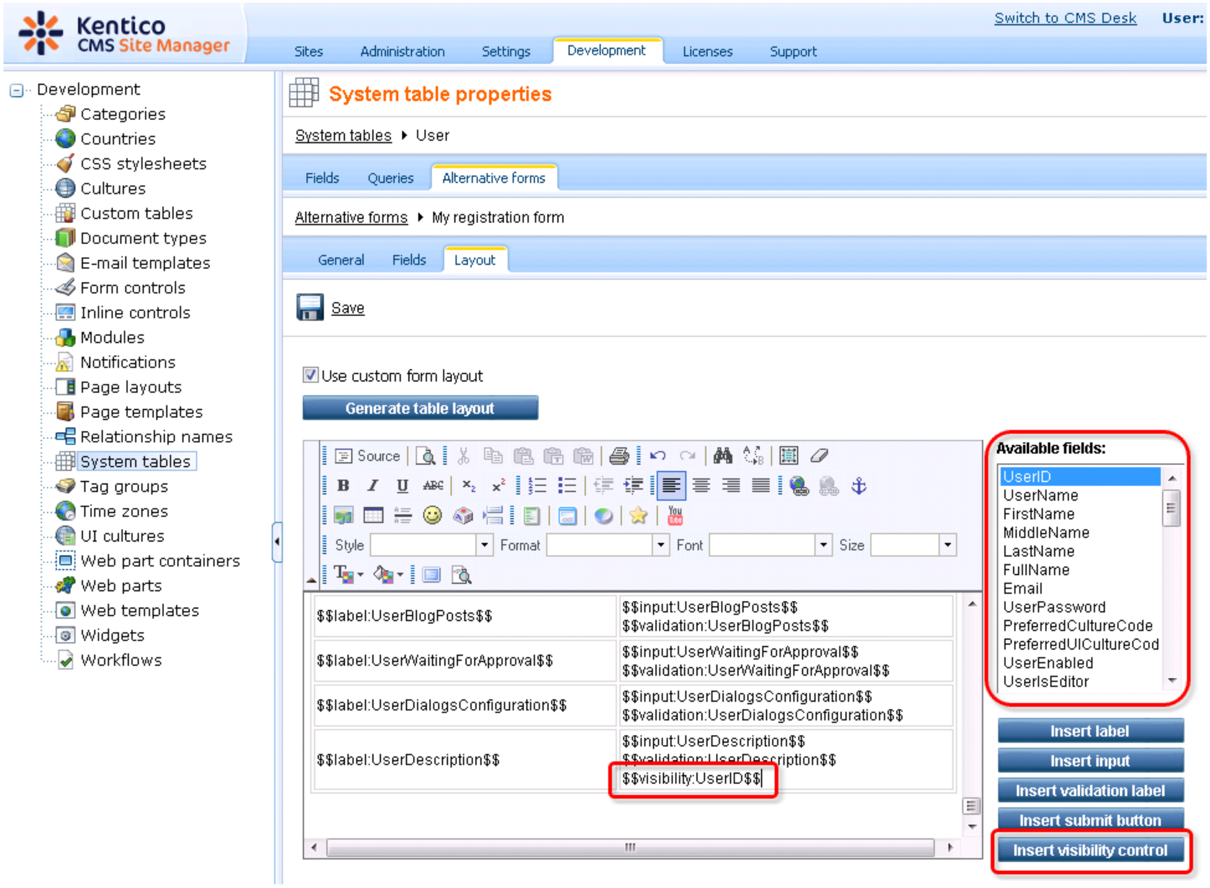
If you want to define a custom form layout and use the visibility drop-down list for some field, you have to do the following two things:

- the **Allow user to change visibility** check-box must be enabled for each field that you want to use the drop-down list for. If it is not enabled, the drop-down list will not be functional.
- add the drop-down list manually to the form, using the **Insert visibility control** button:

1. Go to **Site Manager -> Development -> System tables**. Choose to **Edit** ( ) the **User** system table and switch to the **Alternative forms** tab. Choose to **Edit** ( ) the **Edit profile** alternative form and switch to the **Layout** tab. Check the **Use custom form layout** check-box.

2. Click the **Generate form layout** button. A default form layout will be generated and you can make modifications to it.

3. Enter the visibility controls by placing the cursor to the desired location, selecting the appropriate field and clicking the **Insert visibility control** button. Click **Save** when you are finished.



7.10.10.4 Configuring the web parts

If you want to enable visibility controls in these web parts, you have to add the controls to the appropriate alternative forms and set the following properties of the web parts:

User public profile

- **Form name** - specify the full name of the desired alternative form (*cms.user.DisplayProfile* by default)
- **Apply user's visibility settings** - check this to enable the visibility controls
- **Use visibility settings from form** - if left blank, the form entered in Form name will be used to get visibility settings; if filled, the form specified here will be used (*cms.user.EditProfile* by default);

Custom registration form

- **Alternative form** - specify the full name of the desired alternative form (*cms.user.RegistrationForm* by default)

My account

- **Form name** - specify the full name of the desired alternative form (*cms.user.EditProfile* by default)
- **Allow user to edit field visibility** - check to enable the visibility controls

7.10.11 Authentication

7.10.11.1 Authentication overview

The system supports both forms and Windows authentication. The **forms authentication** stores user names and passwords in the database and requires users to log on. The **Windows authentication** gets user identity from the network credentials and creates the user automatically in the database, including the user's roles (if they exist in the CMS database).

Accessing current user information in code

When the user is authenticated, a CMS.CMSHelper.CurrentUserInfo object representing the current user is stored in the session variable CMSCurrentUser and is accessible through the CMSHelper.CMSContext.CurrentUser property. All operations after authentication then use the user profile and user roles from this object.

[C#]

```
// getting user name
string userName = CMS.CMSHelper.CMSContext.CurrentUser.UserName;
```

Configuring forms authentication

The forms authentication is configured by default. It uses the standard ASP.NET forms authentication and its settings. You can find the settings in the web.config file:

```
<system.web>
  <authentication mode="Forms">
    <forms loginUrl="CMSPages/logon.aspx" defaultUrl="Default.aspx" name=".
ASPXFORMSAUTH" timeout="60000" slidingExpiration="true" />
  </authentication>
<system.web>
```

Password Encryption

You can also configure the **password encryption** in **Site Manager -> Settings -> Security**, in the **Password format** key. By default, the passwords are stored in plain text. If you want to use SHA1 hash format, please set the value to **SHA1**. Then, you need to set all password again so that they are stored in the new format - thus, it's recommended that you make the change after the installation, before you create user accounts.

Membership provider and ASP.NET 2.0 Membership support

Kentico CMS contains an ASP.NET 2.0 Membership provider for the Kentico CMS user database. It means you can use ASP.NET 2.0 Membership API and controls, such as Login control. However, Kentico CMS uses its own user information database instead of the ASP.NET 2.0 Membership tables.

Forgotten administrator's password

If you happen to lost your administrator's password, you can still recover from this:

- If you're using plain text passwords, you can change the password directly in the CMS_User table.
- If you're using hashed passwords, you need to set the password in the CMS_User table to an empty string. Then, you can sign in to **Site Manager** with an empty password and change the password.

Configuring Windows authentication

Please see the chapter Configuring Windows authentication.

Configuring custom authentication

If you want to use user and role information from an external source (such as a custom database), you need to configure the system as described in chapter Configuring custom authentication.

7.10.11.2 Windows authentication (Active Directory)

7.10.11.2.1 Configuring Windows authentication (Active Directory)

Kentico CMS supports Windows integrated authentication. It means that when a user signs in to a Windows domain, Kentico CMS automatically recognizes their identity without requiring a user name and password.

Moreover, Kentico CMS is able to automatically import the authenticated users from domain (Active Directory) into the user database, including their roles.

Configuration

1. Before you configure the application for Windows authentication, you need to create a user account that will be the same as your current domain name and assign this user account with administrator permissions. This will allow you to access all features as an administrator once you sign in using Windows authentication.

2. Sign in as an Administrator to the **Site Manager** and go to **Administration -> Users**. Create a new user with following values:

- **User name:** your domain user name in format *domain-username*, example: *office-johns*
- **Full name:** your full name

Click **OK**.

3. On the **General** tab, set the following values:

- **Is global administrator:** yes
- **Is external user:** yes
- **Is domain user:** yes

Click **OK**.

4. Now you can switch the application to the Windows authentication mode. Edit the web.config file of the web project and change the following line:

```
<authentication mode="Forms">
```

to:

```
<authentication mode="Windows">
```

5. Optional: when using Windows authentication, you may also want to have the following settings in your *web.config* file so that the windows authentication is required for access to the live site:

```
<location path="">
  <system.web>
    <authorization>
      <deny users="?"/>
    </authorization>
  </system.web>
</location>
```

6. Save the modified web.config file. Close all browsers with Kentico CMS and open the website in a new browser. Try to go to **<web project>\cmssitemanager** to make sure you are recognized as a global administrator.

With this configuration, when an authenticated user comes to the site, their user account is created in Kentico CMS database automatically and their domain groups are imported as roles into Kentico CMS database. It means that the users and roles are not imported on some regular basis, but they are imported when the user comes to Kentico CMS website.

If you are experiencing the 401 error on Windows 7 or Windows Server 2007, learn the solution to the problem here.

Sign out button missing with Windows authentication

When Windows authentication is enabled, the **Sign out** button in the top right corner of **CMS Desk** or **Site Manager** is not displayed. The same applies to the live site, where the sign out link is not displayed in all web parts that can be used to sign out.

Forbidden characters replacement on Active Directory import

When importing users and roles, forbidden characters in their names are replaced by the character defined in **Site Manager -> Settings -> URLs -> Forbidden characters replacement**.

Dash "-" is the default value and therefore it is used in this example (*domain-username* instead of *domain\username*). If you are using a different character, please change the entered user name accordingly.

You can override this setting by using the following keys in the *AppSettings* section of your *web.config* file. In both cases, the value must be exactly one character which will be used as the replacement character:

- *<add key="CMSForbiddenUserNameCharactersReplacement" value="-" />*
- *<add key="CMSForbiddenRoleNameCharactersReplacement" value="-" />*

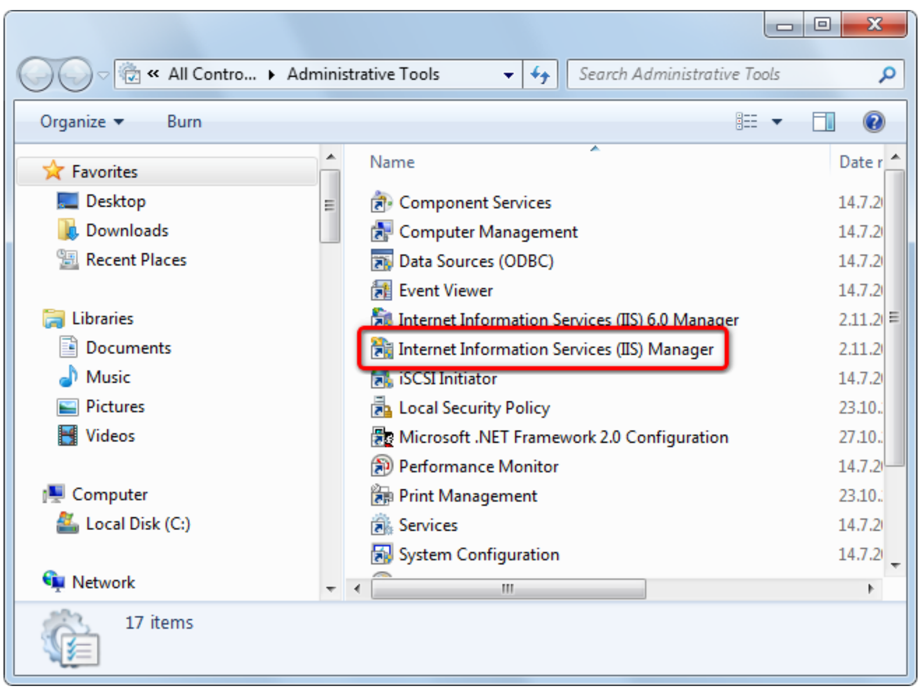
If you want to achieve the same functionality as in **older versions** of Kentico CMS (**office\username**), forbidden characters replacement can be turned off completely using the following two keys. This may cause problems when using wildcard URLs with user names in the wildcard part and is therefore not recommended.

- *<add key="CMSEnsureSafeUserNames" value="false" />*
- *<add key="CMSEnsureSafeRoleNames" value="false" />*

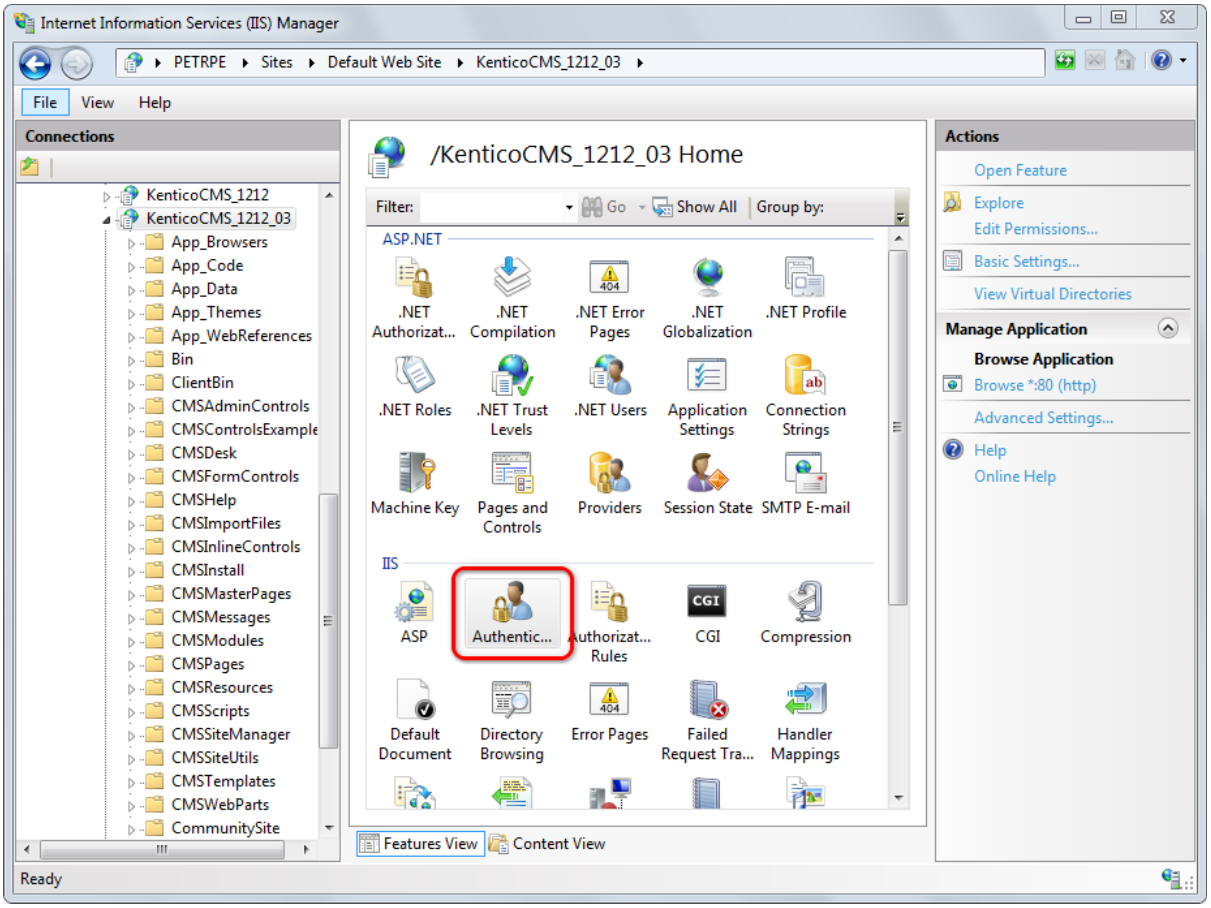
7.10.11.2.2 Windows authentification on Windows 7/2008 R2/Vista (IIS7 or higher)

If you are experiencing the **401 error** with Windows authentication on **Windows 7**, **Windows Server 2008 R2** or **Windows Vista**, you have to **set up your IIS** the following way:

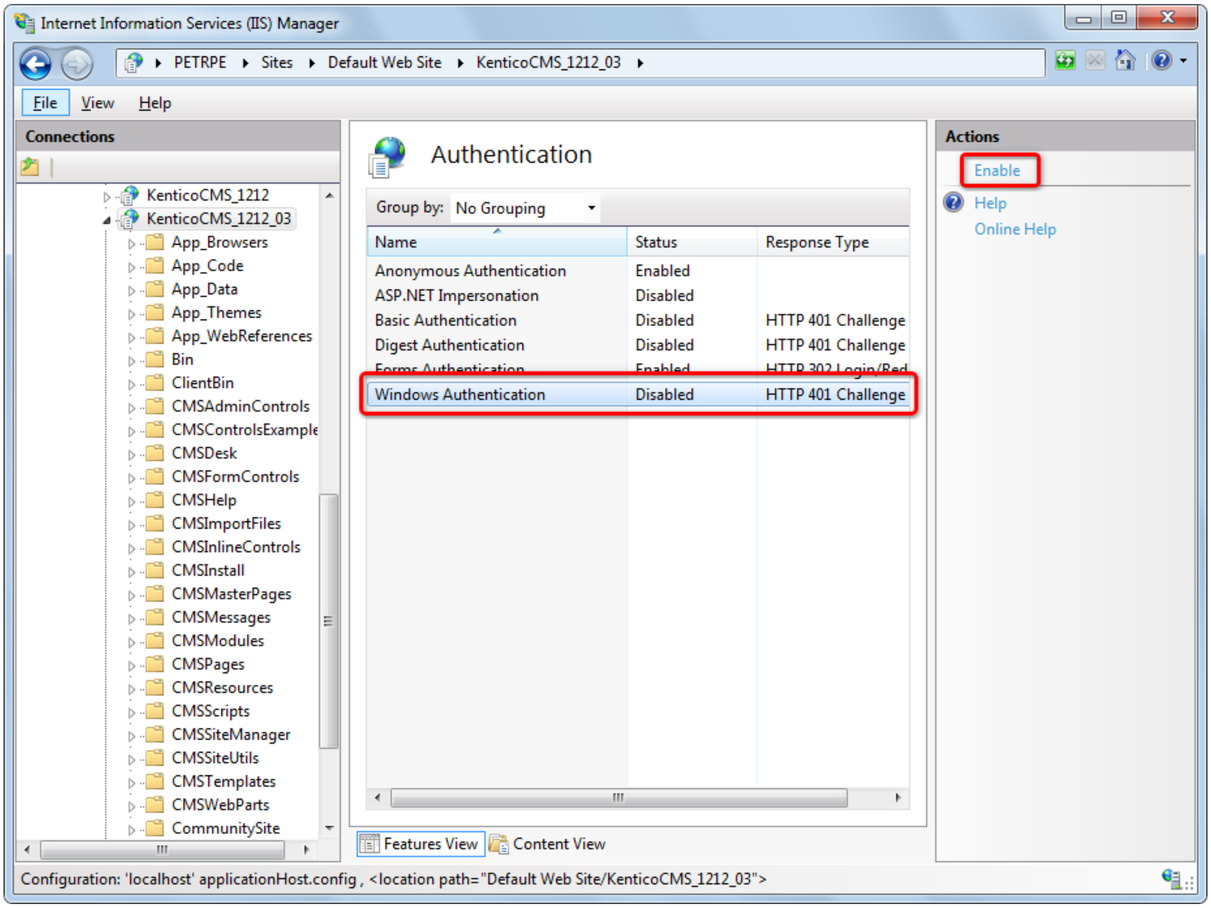
1. Go to **Start** -> **Control Panel** -> **Administrative Tools** and start the **Internet Information Services (IIS) Manager**.



2. Locate and select your site in IIS tree and click on the **Authentication** icon.



3. Enable **Windows Authentication** by clicking on the **Enable** link in the **Actions** menu.



Windows Authentication missing in the list

The default installation of Windows 7, Windows 2008 R8 and Windows Vista does not contain Windows Authentication by default. If it is missing in the list is step 3 above, you need to install it.

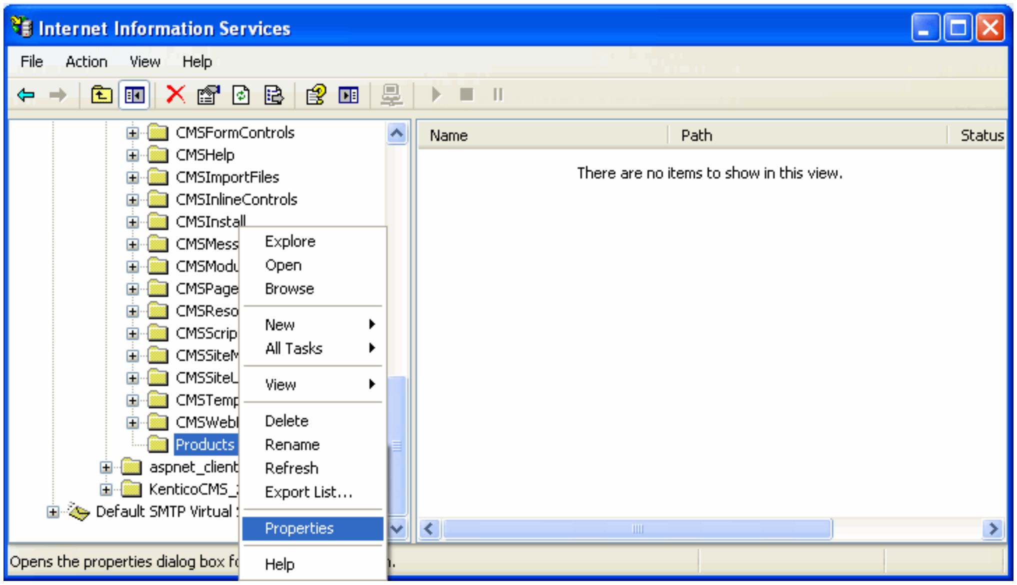
To do it, go to **Control Panel -> Programs and Features -> Turn windows features on or off**. Scroll down to *IIS*, expand all of the nodes to find the *Security* node and under it, check the Windows Authentication check-box. Click **OK** to save the settings.

7.10.11.2.3 Securing a w eb site section using Window s authentication

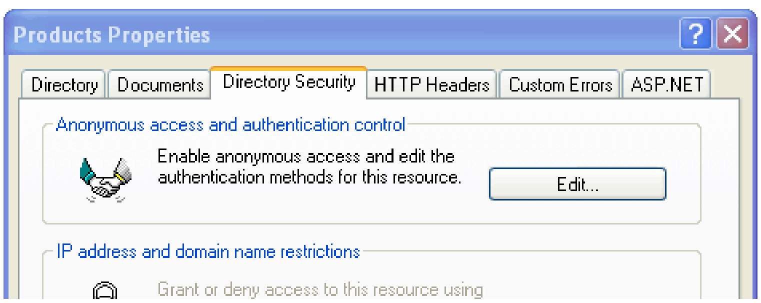
It is also possible to secure only a certain section of your website using Windows authentication. In the following example, you will learn how to set the **Products** section of our sample **Corporate site** to be secured by the Windows authentication:

1. Locate your web project on the disk (typically *c:\Inetpub\wwwroot\<web project>*). Create a new directory in your web project's folder and give it the same name as the filename in the document's URL. In this case, the filename is *Products.aspx*, so we will create a folder named **Products**.

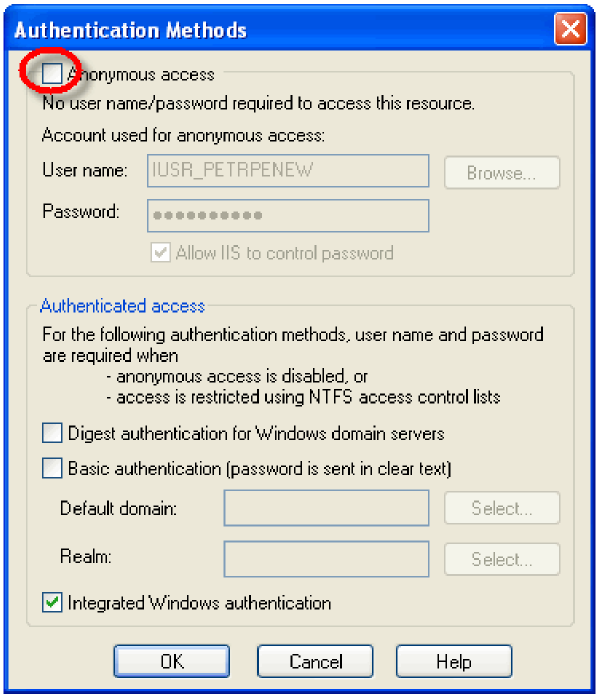
2. Open the IIS and locate the directory in the tree. Right click it and choose **Properties**.



3. Switch to the **Directory security** tab and click the **Edit** button placed in the **Anonymous access and authentication control** section.



4. Uncheck the **Anonymous access** check-box and click **OK**. Check the same setting for the root of the web and make sure that Anonymous access is allowed for it.

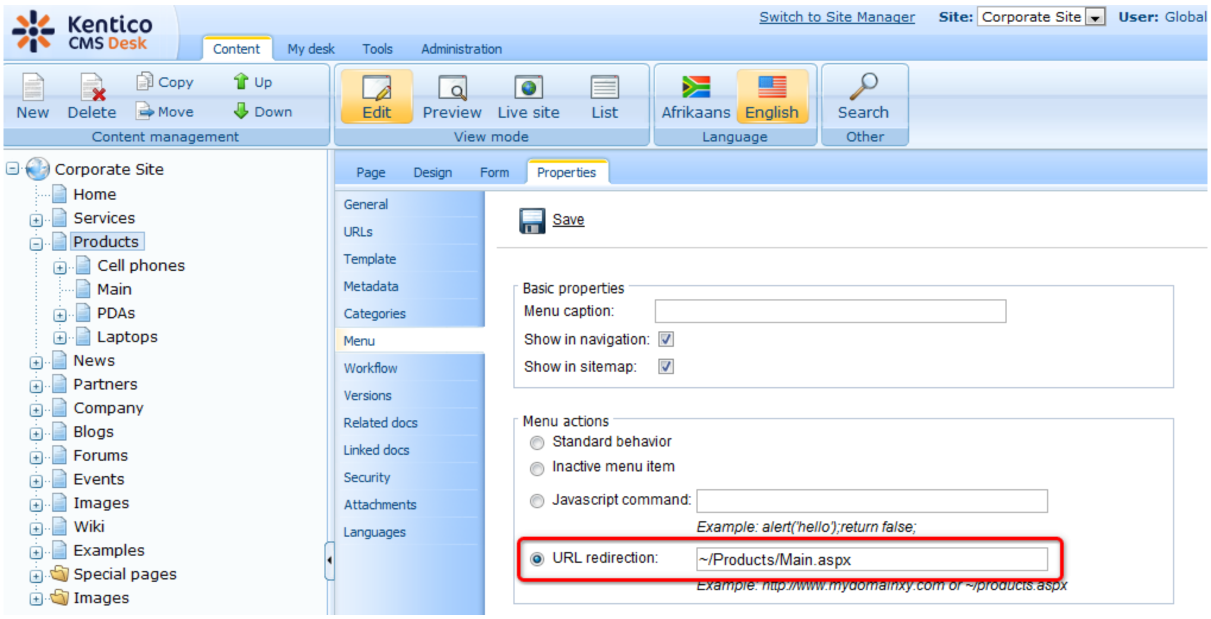


5. Open the web.config file of your web project and change value of the **mode** attribute of the **authentication** tag to **Windows**. Also find the section marked with **Windows authentication BEGIN** and change the path parameter of the location tag to the name of the created directory, which will be **Products** in our case:

```
...
<authentication mode="Windows">

...

<!-- Windows authentication BEGIN -->
<location path="Products">
  <system.web>
    <authorization>
      <deny users="?"/>
      <allow users="*"/>
    </authorization>
  </system.web>
</location>
<!-- Windows authentication END -->
```

6. The authentication is now configured. If you try to access any of the menu items placed under the **Products** section, **Windows authentication** will be required. However, if you also want the authentication to be required for the **Products** main page (which is obviously not located under itself, hence requires no authentication now), you will have to use the following workaround.

Create a new page under the Products section, give it the same content as the main page has got and redirect the Products link in the menu to this new page. Because the new page is located under the Products section, windows authentication will be required for it.



7.10.11.2.4 Active Directory Import Utility

7.10.11.2.4.1 AD Import Utility overview

Introduction

Kentico CMS Active Directory Import Utility is a standalone Windows application which allows importing of users and groups (roles) from Active Directory (AD) into Kentico CMS and assigning users to roles depending on AD settings. The application also provides the possibility of updating already imported users and roles so that their properties are the same as in the current AD.

What it can do?

- Import users from AD into Kentico CMS.
- Import roles (groups) from AD into Kentico CMS.
- Assign users to appropriate roles based on AD settings.
- Update already imported users and roles according to current AD.

What it can't do?

- Import from multiple ADs or domains at once.
- Import the tree structure of roles, since Kentico CMS does not support hierarchical roles.
- Since there is no hierarchy in the CMS roles, import cannot keep the tree structure of AD groups.

Terminology

- **Import profile** – XML file with import settings; this file can be created using the wizard mode, or even written manually; it is necessary to have an import profile prepared when you want to use the console mode of the tool
- **SAM Account Name -** logon name used to support clients and servers on older versions of the

operating system, such as Windows NT 4.0, Windows 95, Windows 98, and LAN Manager.

- **UPN (User Principal Name)** - Internet-style login name for a user. It is based on the RFC 822 standard. The UPN is shorter than the distinguished name and easier to remember. By convention, the name should map to the user's e-mail name. The value set for this attribute is equal to the length of the user's ID and the domain name. (Sample UPN: username@subdomain.domain.tld)
- **Role or Group** - these two terms have almost identical meaning, while ‚group' comes from AD terminology and ‚role' is used in Kentico CMS

Requirements

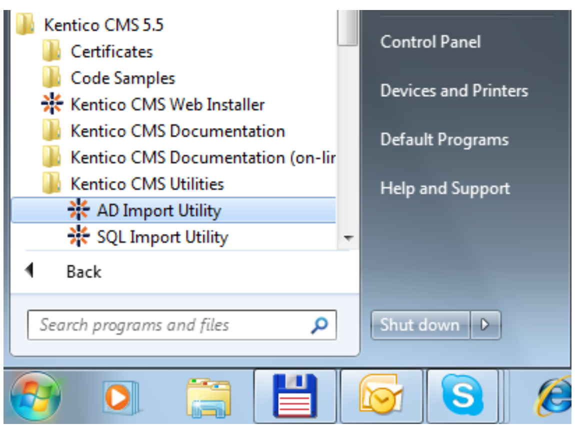
- **Kentico CMS 5.0** or higher
- **Ultimate edition** or any edition with the **Advanced package**

Launching the utility

There are two ways how the utility can be launched. You will typically use the AD import wizard, which is described in the Using the wizard chapter. If you want to schedule AD import to be performed on a regular basis, you may utilize the second option - launching AD import from the command line. This approach is described in the AD import from command line chapter.

7.10.11.2.4.2 Using the wizard

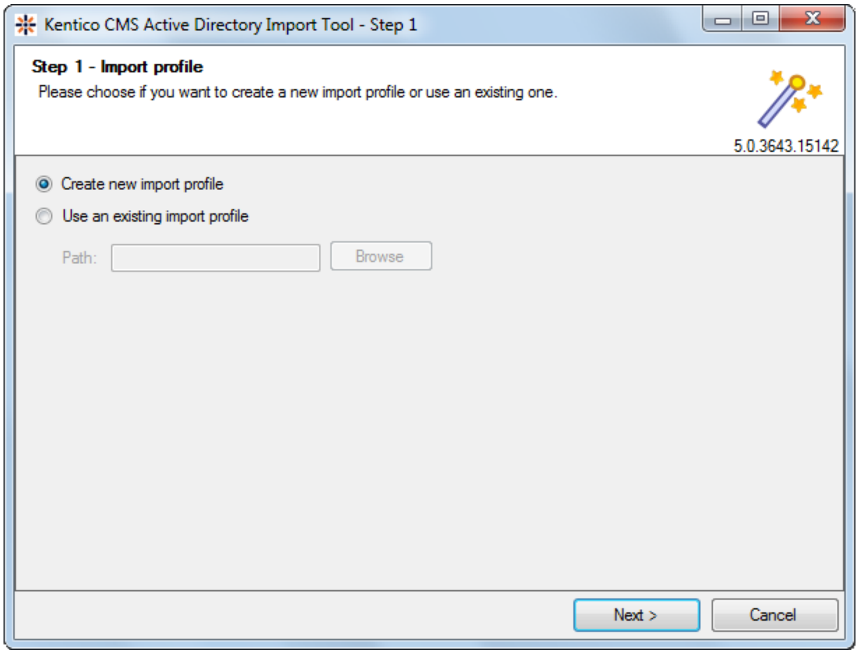
The AD Import Utility wizard can be executed either from **Start menu -> All programs -> Kentico CMS <version number> -> Kentico CMS Utilities**, or by launching the **ADImport.exe** file located in **<Kentico CMS installation folder>\ADImport** (typically *c:\Program Files (x86)\KenticoCMS\5.5 \ADImport*).



The following text describes particular steps of the wizard:

Step 1 – Import profile settings

In the first step, you need to choose if you want to create a new import profile or use an existing XML profile. If you select an existing profile, values will be pre-filled in the following steps based on the profile settings.



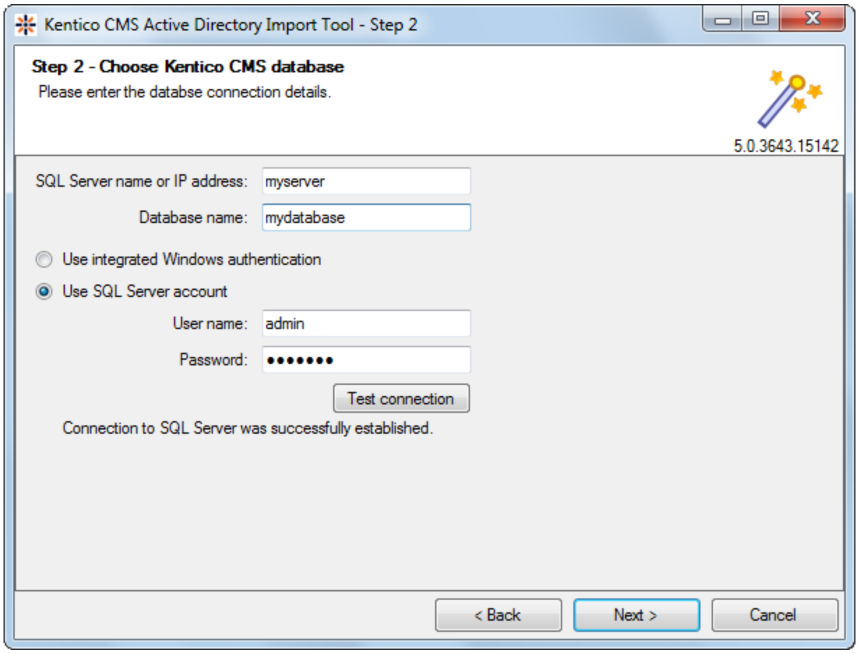
Step 2 – Kentico CMS DB Setup

In the second step, specify the target database of the CMS where the users and roles will be imported:

- **SQL Server name or IP address**: name or IP address of the server where the target database is stored
- **Database name**: name of the target database

- **Use integrated Windows authentication**: choose this option if you want to log on to the server using Windows authentication
- **Use SQL Server account**: choose this option if you want to log on to the server using credetials filled in the fields below

It is a good idea to test the specified connection using the **Test connection** button before proceeding to the next step.

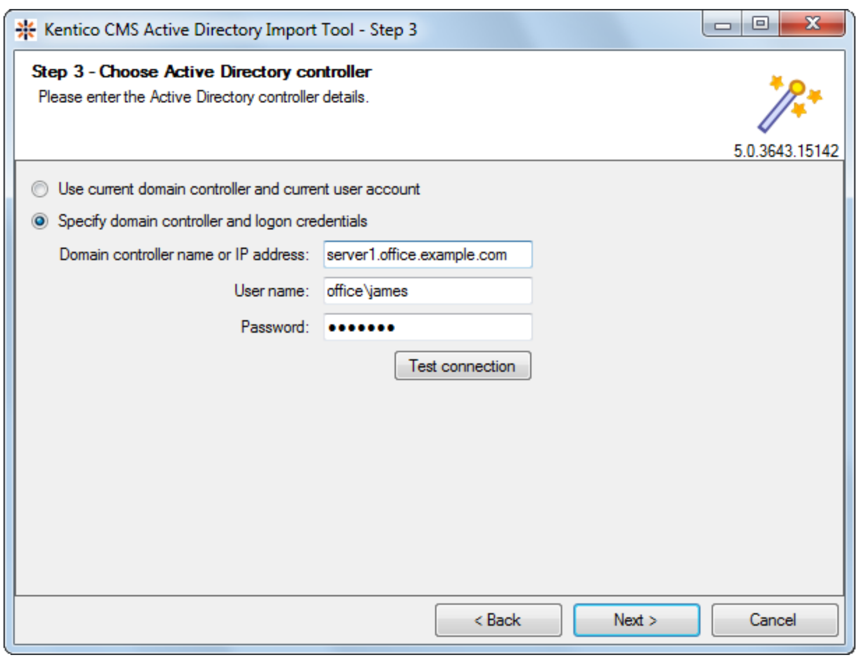


Step 3 – Active directory connection

In the third step, specify the source AD's domain controller. You have two options:

- **Use current user account**: uses the domain where the current user belongs
- **Specify domain controller and logon credentials**: if you choose this option, you can enter the logon details manually into the fields below

Here again, it is recommended to test the specified connection using the **Test connection** button.



Step 4 – Import settings

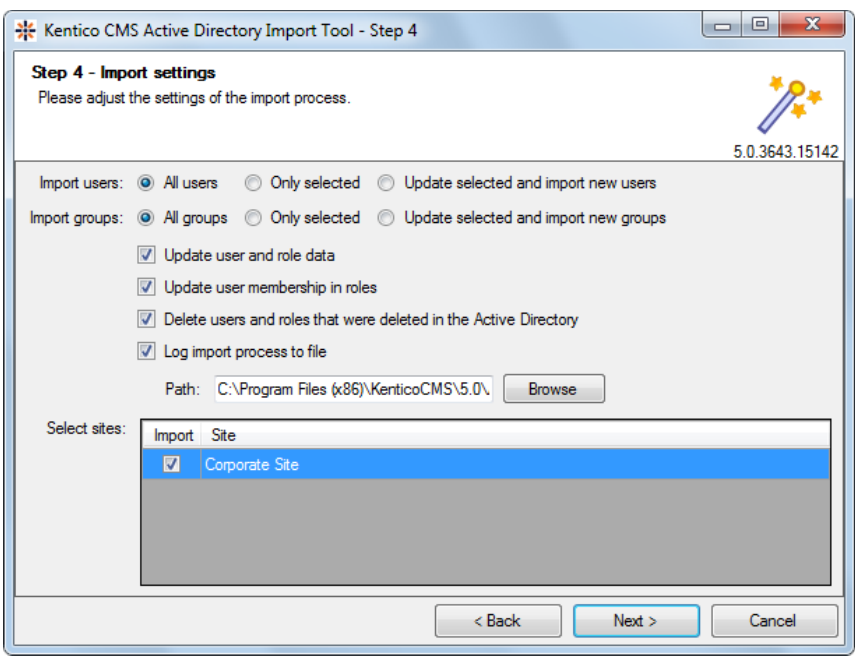
In this step, you can adjust some general settings of the import process:

- **Import users/groups**: determines which users or groups (roles) will be pre-selected in Step 6, you have the following options:
  - **All**: all users or groups will be pre-selected
  - **Only selected**: when using an existing import profile, selection stored in the profile will be used; otherwise, nothing will be pre-selected
  - **Only selected and new**: same as above, while new users and groups will be selected as well

- **Update user and role data**: if enabled, properties of users and roles already imported from the AD will be updated in the CMS based on the current values in AD
- **Update user membership in roles**: if enabled, membership of users imported from the AD will be updated in the CMS based on the current membership settings in AD
- **Delete users and roles that were deleted in the Active directory**: if enabled, users who were previously imported from the Active Directory but were deleted on the source server since then will be deleted in the CMS
- **Log import process to file**: if enabled, you can specify a file where the import log will be stored

- **Select sites**: choose the sites to which the imported users and roles will be assigned

Please note: If you do not choose any site in this step, the rest of the wizard will leave out steps related to roles (groups) import. This happens because roles are site-related in Kentico CMS (i.e. each role must belong to some site).

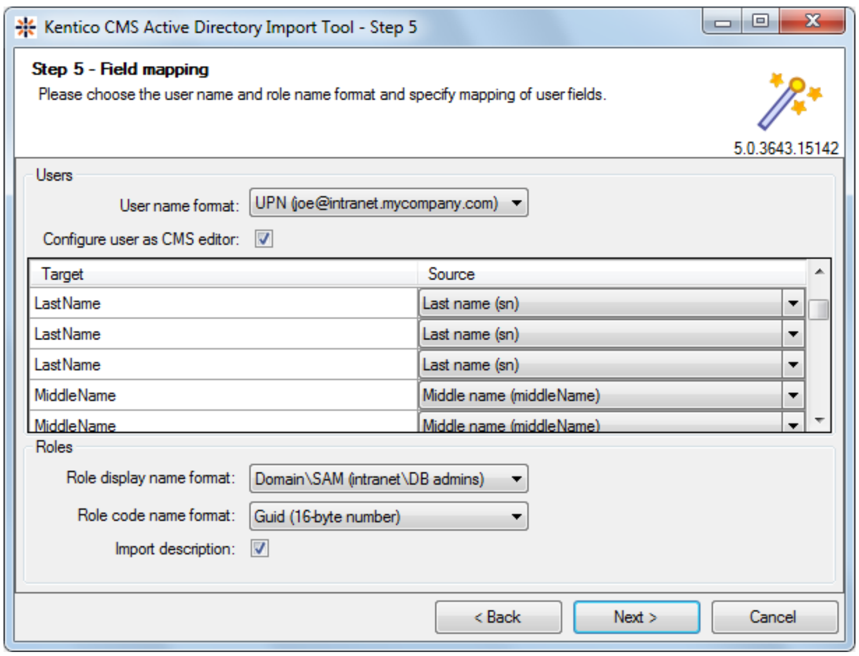


Step 5 – Import properties

In this step, you are asked to define user name and role name format and to bind AD user properties to CMS user properties. The following options can be defined:

- **User name format**: choose one of the three possilbe formats:
  - Domain\SAM (e.g. intranet\joe)
  - SAM account name (e.g. joe)
  - UPN (joe@intranet.mycompany.com)
- **Configure user as CMS editor**: if enabled, all imported users will have the Is editor option enabled (this option is located in Administration -> Users -> user edit -> General)
- **Target/Source**: you can choose which properties from AD (Source) will be mapped to particular properties of the CMS_User role

- **Role display name format**: choose one of the two possible formats:
  - Domain\SAM (intranet\DB Admins)
  - SAM (DB Admins)
- **Role code name format**: choose one of the following formats:
  - Domain\SAM (intranet\DB Admins)
  - SAM (DB Admins)
  - Guid (16-byte number)
- **Import description**: indicates if role description should be imported from the AD



Step 6 – Select users & groups to be imported

In the sixth step, you can select which roles and users will be imported. It will be possible to adjust the settings made here in the following two steps.

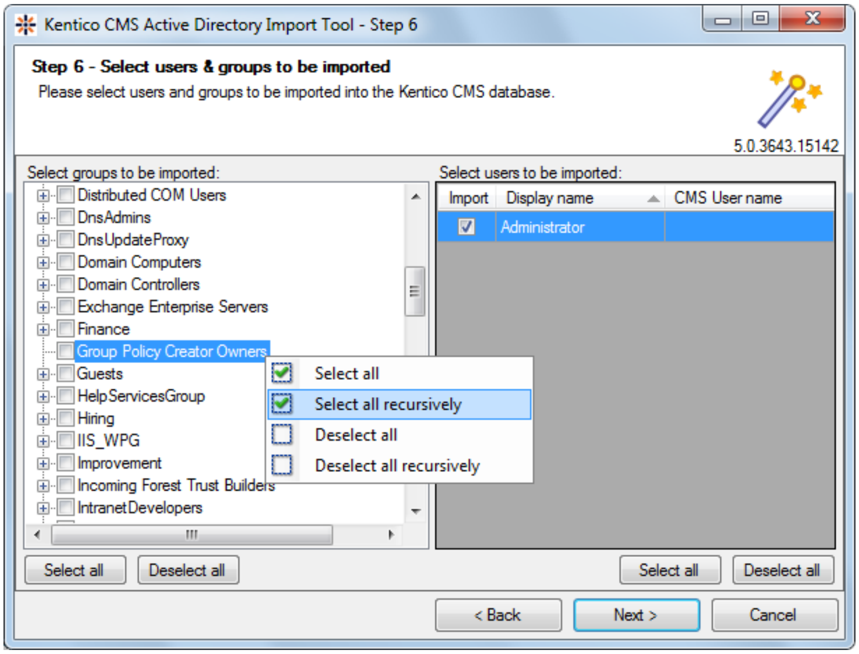
On the left, you can see all groups (roles) found on the source server. If you select a group, its members are displayed in the list on the right. You can define which users and roles will be imported using the appropriate check-boxes.

By right-clicking a group, you can display a context menu with the following actions:

- **Select all**: selects all child groups directly under the selected group
- **Select all recursively**: selects all child groups under the selected group until the last level
- **Deselect all**: selects all groups directly under the selected group

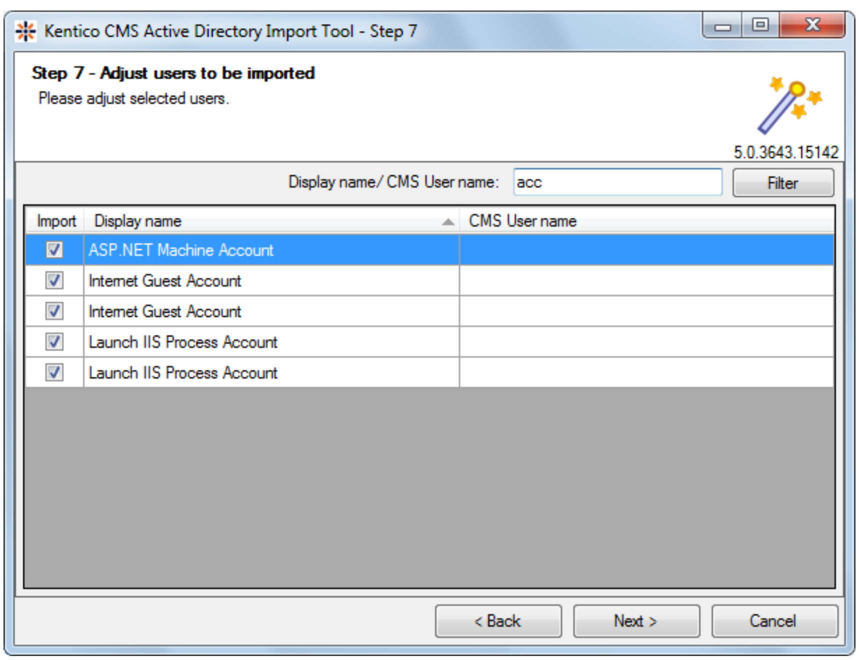
- **Deselect all recursively**: selects all group under the selected group until the last level

All users in a role or all roles can be selected or deselected in one click using the **Select all** and **Deselect all** buttons.



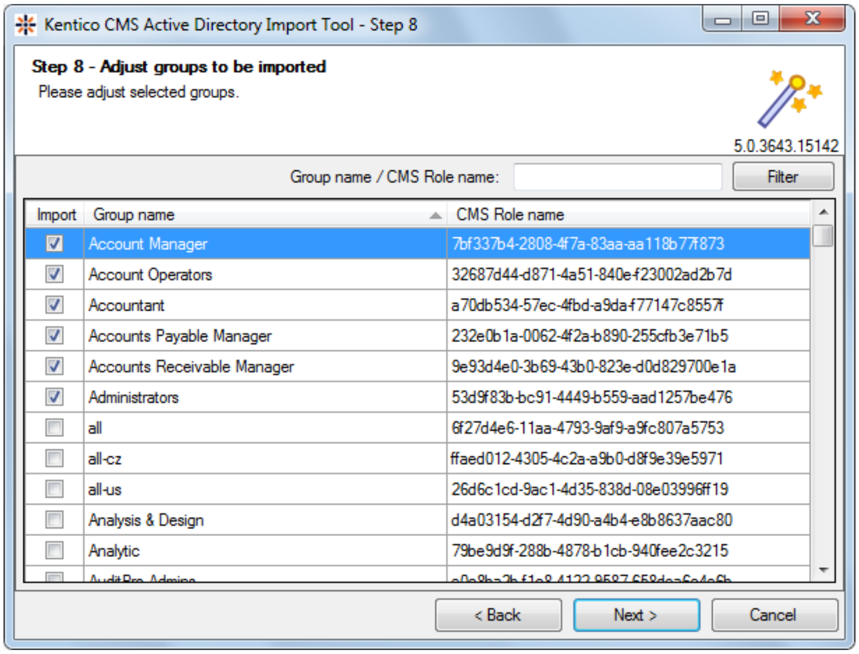
Step 7 – Adjust users to be imported

In this step, you can adjust the users to be imported. Users are selected based on settings made in the previous step, while you can adjust the selection using the check-boxes.
You can also filter the listed users by *Display name* and *User name* using the filter above the list.



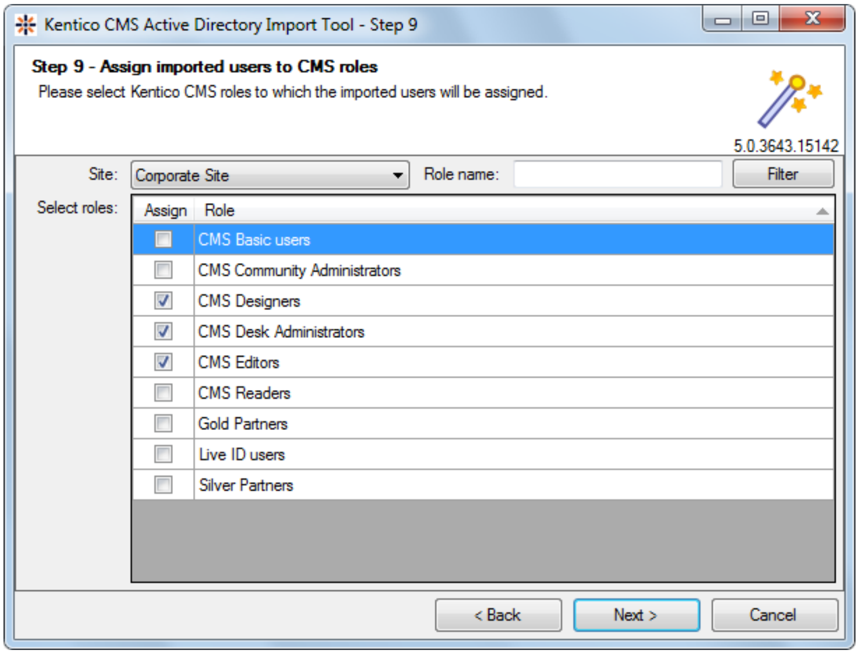
Step 8 – Adjust groups to be imported

This step is similar to the previous one, while groups (roles) to be imported can be adjusted here using the check-boxes.
Listed groups can be filtered by *Group name* using the filter above the list.



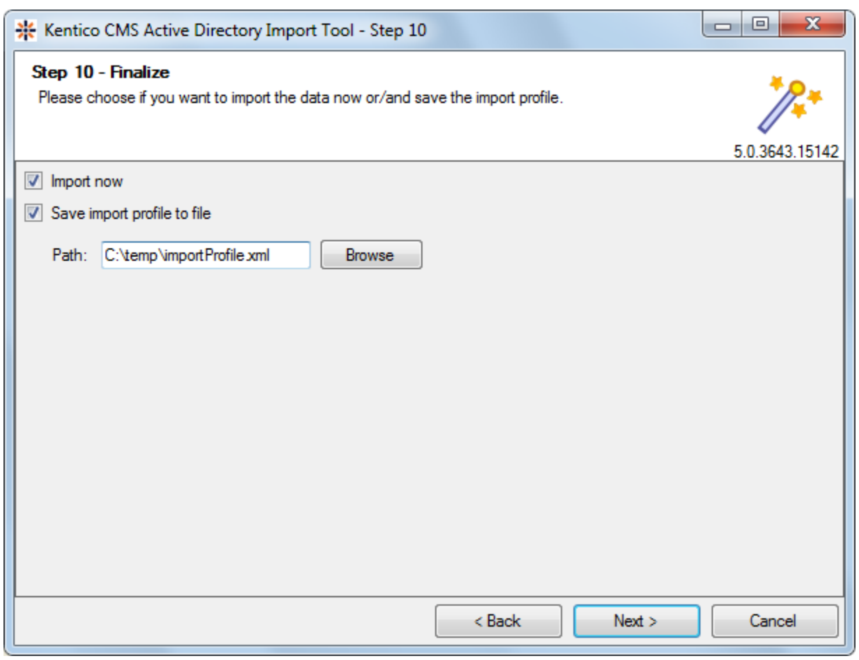
Step 9 – Assign to CMS roles

In the sixth step, you can select roles to which the imported users will be assigned. If you are importing to multiple sites, you first need to choose the site whose roles should be displayed using the **Site** drop-down.



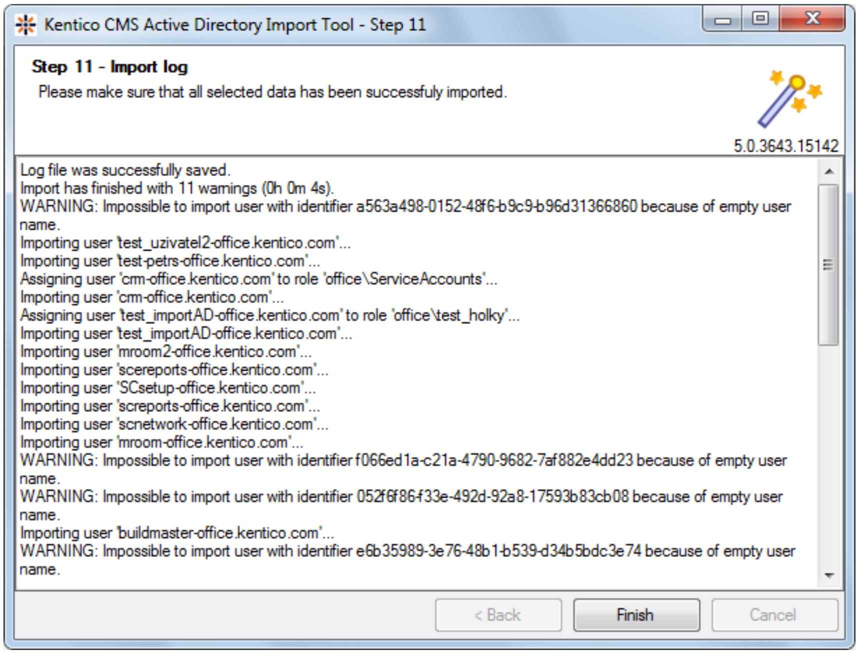
Step 10 - Finalize

Now you have your import profile configured. You can execute it immediately, save it into a file or perform both of these actions, depending on which of the **Import now** and **Save import profile to file** check-boxes is enabled.



Step 11 – Import log

The last step displays an import log, showing the progress of the import process. When the import finishes, you can close the wizard using the **Finish** button.



7.10.11.2.4.3 AD import from command line

Apart from the wizard described in the Using the wizard chapter, the AD Import Utility can also be launched from windows command line. It is achievable by executing the **ADImport.exe** file located in **<Kentico CMS installation folder>\ADImport** (typically *c:\Program Files (x86)\KenticoCMS\5.5 \ADImport*) using a special syntax.

You can launch the utility with the **-h** parameter, which displays help on using the utility from the command line:

```
ADImport -h
```

To perform the actual import, you need to have an import profile created via the wizard. With the import profile prepared, you need to execute the utility using the **ADImport /profile <profile file name>** syntax. When specifying the profile file name, both an absolute and a relative path can be used. Please make sure that you use proper quotation when entering an absolute path containing special characters (e.g. blank spaces).

```
ADImport /profile my_profile.xml
ADImport /profile "C:\Temp\AD Import\my_profile.xml"
```

After executing the command, users or groups from Active Directory will be imported to your Kentico CMS instance based on settings contained in the specified import profile.

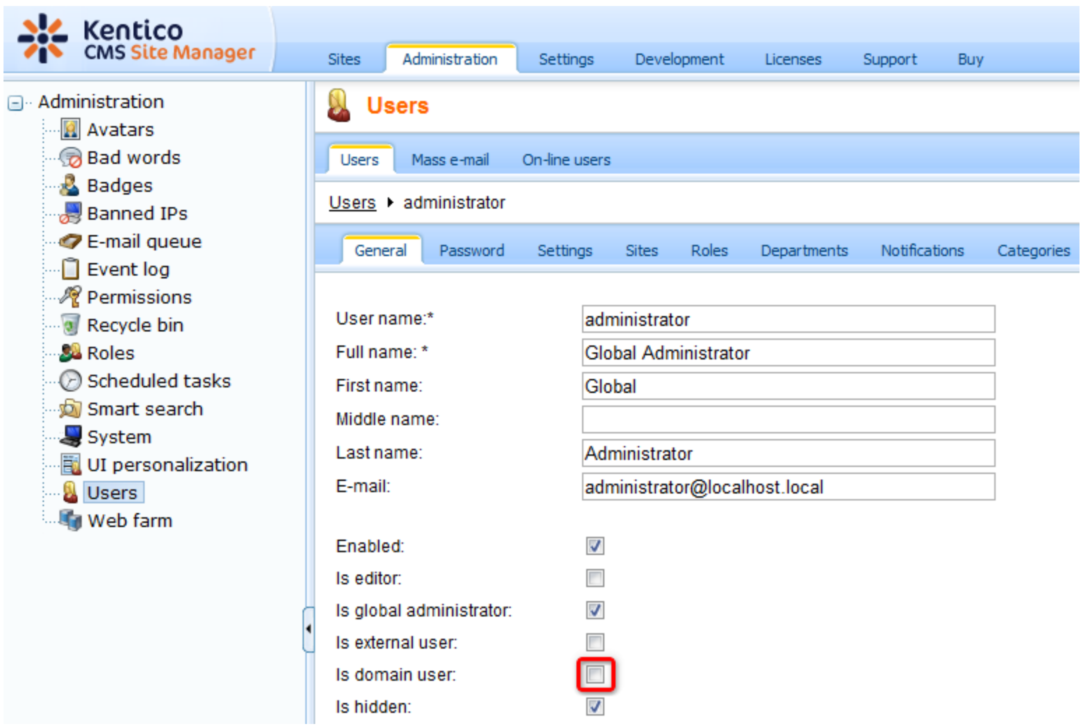
7.10.11.2.4.4 How to recognize imported users and roles

In the CMS, you can recognize users imported from AD by the **Is domain user** check-box on a user's **General** tab, as you can see in the screenshot below.

When editing roles, you can see the **Is domain role** check-box, which has the same meaning for roles.

These check-boxes reflect the values of the following Boolean fields in the database tables:

- **CMS_User -> UserIsDomain**
- **CMS_Role -> RoleIs Domain**



7.10.11.3 Configuring mixed mode authentication

Mixed mode authentication enables users to sign in to your website using both Windows authentication and standard forms authentication at the same time.

To enable this authentication mode, you have to modify your *web.config* file so that the **connectionStrings**, **membership** and **roleManager** sections are the same as the code sample below:

```
<connectionStrings>
  <add name="CMSADConnectionString" connectionString="<LDAP connection string>" />
</connectionStrings>

<membership defaultProvider="CMSProvider" userIsOnlineTimeWindow="30">
  <providers>
    <clear/>
    <add name="CMSProvider" type="CMS.MembershipProvider.CMSMembershipProvider"
connectionStringName="CMSConnectionString" enablePasswordRetrieval="false"
```

```
enablePasswordReset="true" requiresQuestionAndAnswer="false" requiresUniqueEmail="
true" passwordFormat="Hashed"/>
    <add name="CMSADProvider" type="CMS.MembershipProvider.CMSADMembershipProvider
" connectionStringName="CMSADConnectionString" connectionUsername="username"
connectionPassword="password" />
  </providers>
</membership>

<roleManager defaultProvider="CMSRoleProvider" enabled="true" cacheRolesInCookie="
true" cookieName=".ASPROLES" cookieTimeout="30" cookiePath="/" cookieRequireSSL="
false" cookieSlidingExpiration="true" cookieProtection="All">
  <providers>
    <clear/>
    <add name="CMSRoleProvider" type="CMS.MembershipProvider.CMSRoleProvider"
connectionStringName="CMSConnectionString" applicationName="SampleApplication"
writeExceptionsToEventLog="false"/>
    <add name="CMSADRoleProvider" type="CMS.MembershipProvider.CMSADRoleProvider"
connectionStringName="CMSADConnectionString" connectionUsername="username"
connectionPassword="password" />
  </providers>
</roleManager>
```

The <LDAP connection string> part highlighted in green in the code example above should be replaced with the actual connection string. It should be entered in the format shown below. The first part is the full domain. In the second part, the same domain is divided using the DC parts:

```
LDAP://mydomain.example.com/DC=mydomain,DC=example,DC=com
```

User name and password need to be entered only on Windows XP, in the following format:

- **Username**: <domain name>\user
- **Password**: relevant password

When you have entered this code to your *web.config*, users can log in using their Active Directory user name (without domain) and password, or using their standard Kentico CMS user name and password.

You can also enable users to sign-in using their full Active Directory user name (e.g. *MyName@office.example.com*). For this to work, you have to add the following key to the *AppSettings* section of your *web.config* file:

```
<add key="CMSADDefaultMapUserName" value="userPrincipalName" />
```

7.10.11.4 Integrating authentication with external systems

Kentico CMS allows you to write a custom authentication provider. In this way, the provided user name and password are checked against an external user profile source/authentication source and if the user is successfully authenticated, the user account is automatically created/updated in the Kentico CMS database, without copying the user password.

You can learn more about custom authentication in chapter Security handler (CustomSecurityHandler class).

7.10.11.5 Single sign-on

Single sign-on is supported for the following scenarios:

Forms Authentication

You are using Forms authentication and you need to share user identity across applications that run on the same domain and all of them use the standard ASP.NET 2.0 Forms authentication. You need to ensure that:

1. All applications use the same user database or at least the same user names. You may need to integrate the authentication using a custom security handler.

2. The *web.config* file of all applications uses the same authentication cookie name and the path is set to "/":

```
<authentication mode="Forms">
  <forms name=".ASPXFORMSAUTH" path="/" ...="" />
</authentication>
```

3. The *web.config* file of all applications uses the same machine key. The machine key is not present in the web.config by default. You can generate is using various machine key generators that can be found on the Internet. Once you have a key generated, you can add it to the <*system.web*> section the following way:

```
<system.web>
  ...
  <machineKey validationKey="ABCD0708...." decryptionKey="DDFF8943...." validation
="SHA1" />
  ...
</system.web>
```

4. If your applications run on different sub-domains, such as *www.example.com* and *forums.example.com* , you need to set the domain attribute of the authentication cookie to the main domain so that it's shared across domains:

```
<forms name=".ASPXFORMSAUTH" path="/" domain=".mywebsite.com" ...="" />
```

Windows Authentication

You are using Windows authentication. In this case, the user identity is shared within the Windows domain. No additional configuration is required.

7.10.11.6 Displaying personalized content

Kentico CMS allows you to personalize the displayed content based on the current user.

Personalization in short

1. If you want to customize content displayed to the users, you need to grant or deny the READ permission to these users and turn on the **Check permissions** attribute of the appropriate web parts.

2. When the user is not authenticated, the system uses a special user **Public anonymous User (public)**.

Example: Personalizing the Products menu

In this example, you will learn how to display the Products section only to members of the **Customers** and **Partners** roles and how to display the PDAs category only to the members of the **Partners** role.

1. Sign in to **CMS Desk** as administrator.

2. Go to **Administration -> Roles** and create new roles **Customers** and **Partners**.

3. Go to **Administration -> Users** and create a new user **Customer1** with following values:

- **User name**: Customer1
- **Full name**: Testing Customer
- **Enabled**: true
- **Is editor**: no

Click **OK**. Go to the **Sites** tab and assign the user to Corporate Site. Go to the **Roles** tab and add the user to the **Customers** role.

4. Create another user **Partner1** with following values:

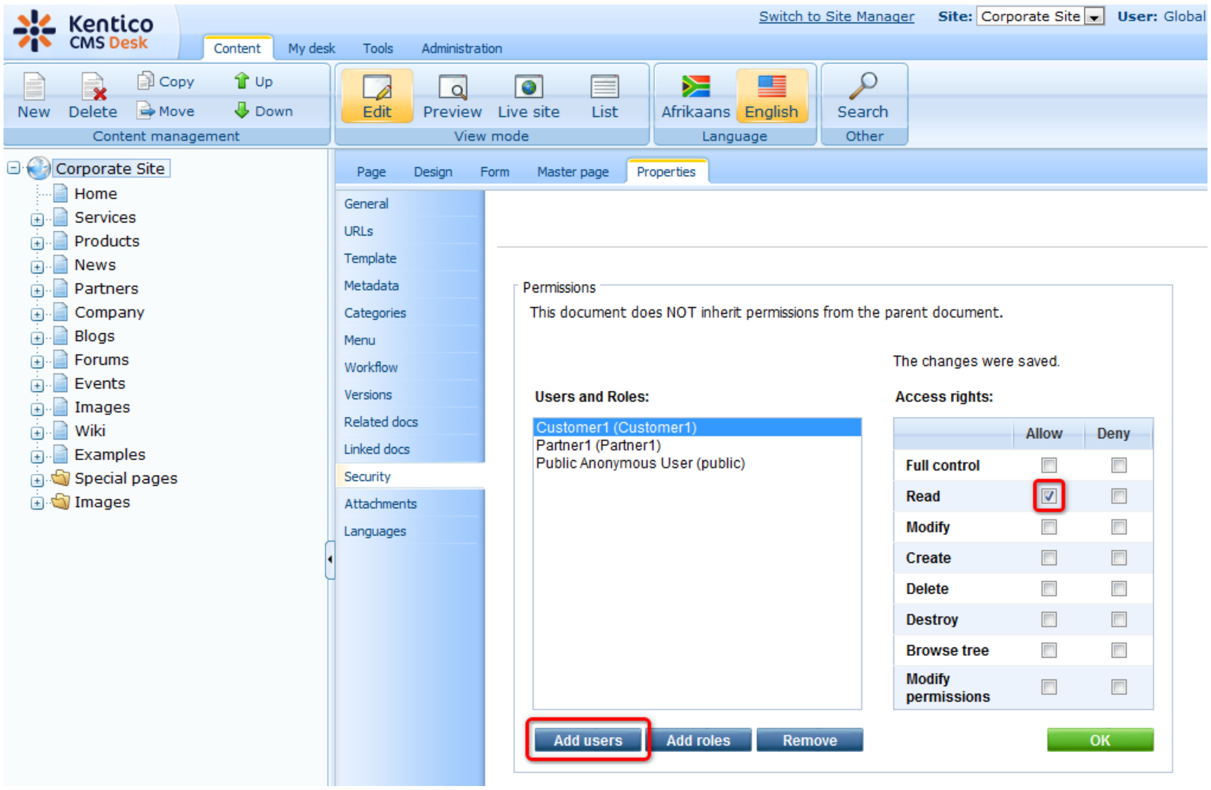
- **User name**: Partner1
- **Full name**: Testing Partner
- **Enabled**: true
- **Is editor**: no

Click **OK**. Go to the **Sites** tab and assign the user to **Corporate Site**. Go to the **Roles** tab and add the user to the **Partners** role.

5. Switch to **CMS Desk -> Content** and click the root of the content tree. Switch to **Properties -> Security**.

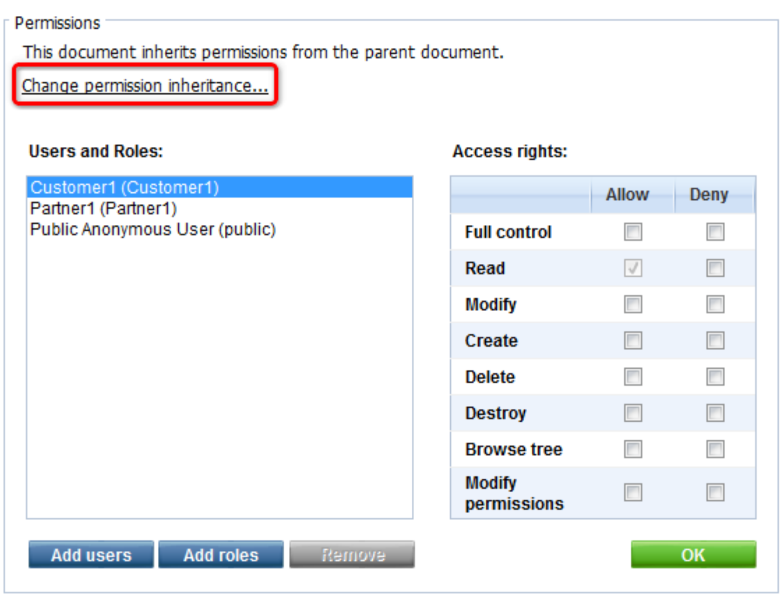
First, we need to grant the **Read** permission for the whole website to the **Public anonymous user (public)**, **Customers** and **Partners** roles.

Click **Add users**, select these the three users and click **OK**. When the users are added, grant the **Read** permissions to them using the check-box (), as you can see in the screenshot below. You need to click **OK** in order to save the settings.



6. Now we will hide the product categories from public users. In the content tree, click **/Products/Cell Phones** and switch to the **Properties -> Security** tab.

The permissions that you configured for the root of the content tree are inherited by this document. We need to break the inheritance. Click the **Change permission inheritance** link. In the following dialog, choose **Break inheritance and copy parent permissions**.



7. Now select the **Public Anonymous User (public)** and deny the **Read** permission.

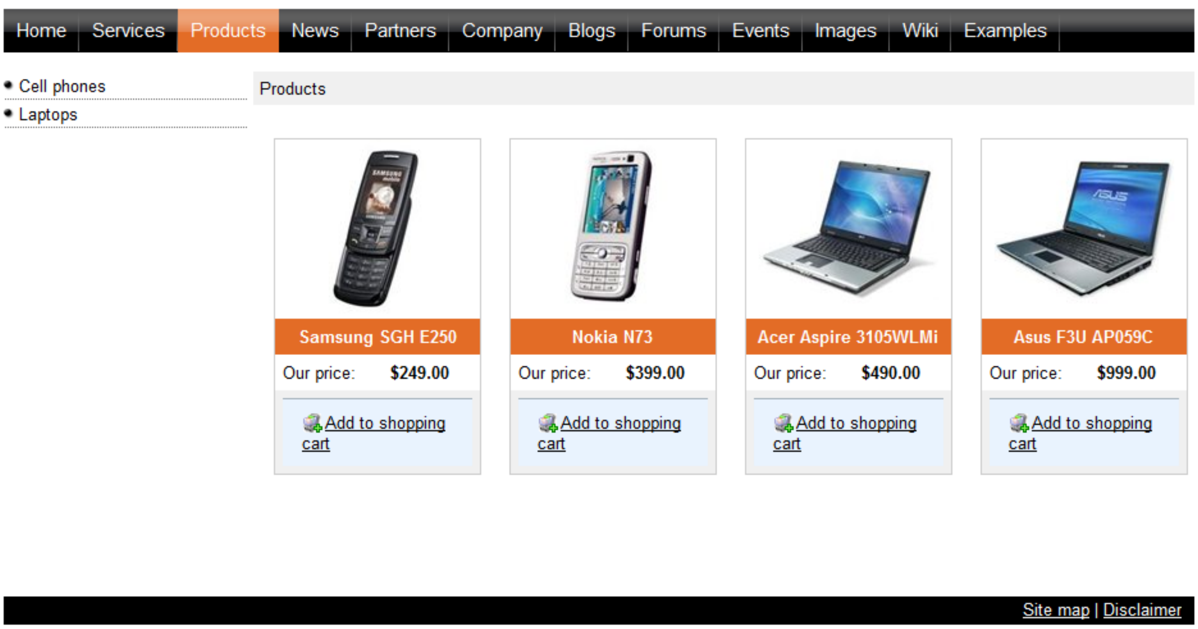
8. Repeat the last two steps for categories **Laptops** and **PDAs**. Deny the **PDAs** category also for the **Customers** role.

9. The permissions are not checked by web parts by default, so we need to configure the web parts so that they check the **Read** permission of the current user.

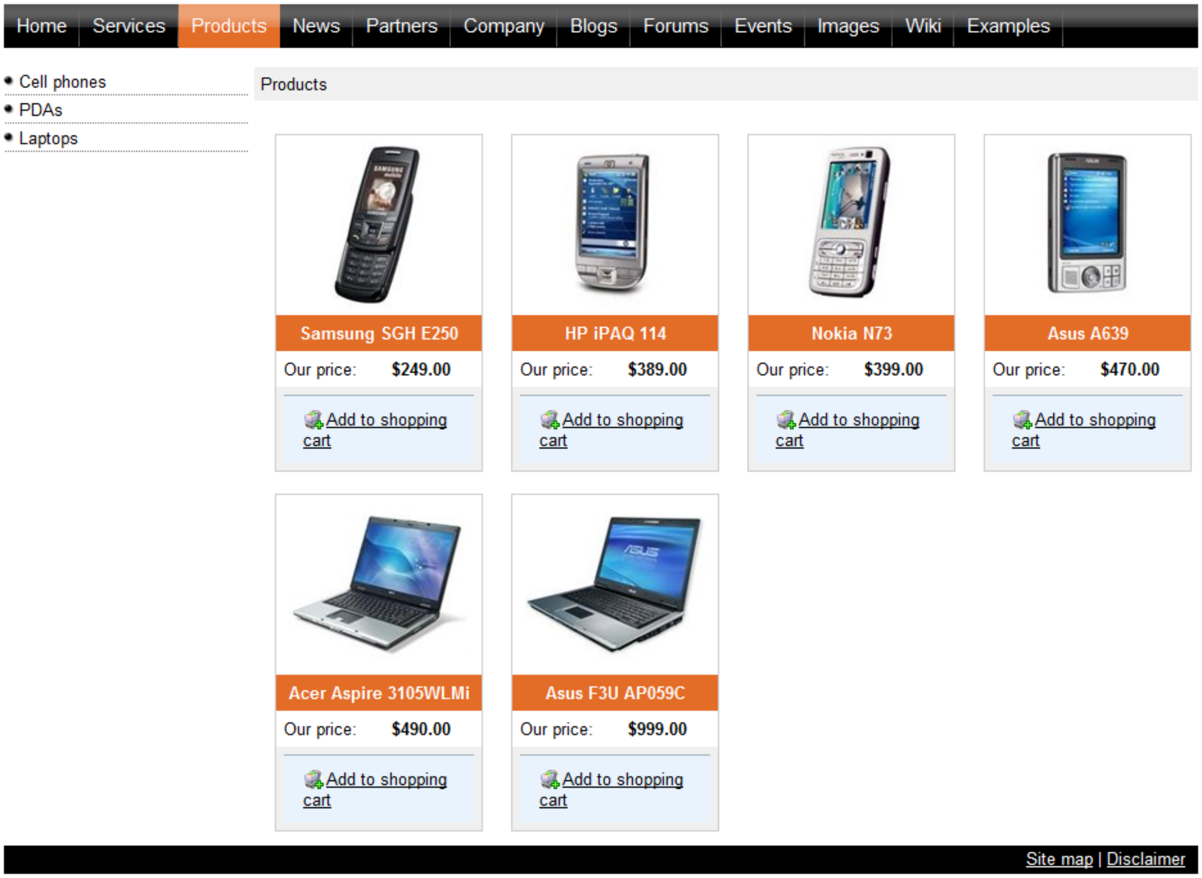
Choose the root in the content tree and click the **Design** tab. Configure the **cmsmenu** web part and set its property **Check permissions** to true (☑). Click **OK**. Repeat the same for the **LeftTreeMenu** and **ProductDataList** web parts on the **/Products** page.

10. Sign out. If you mouse-over the **Products** menu item, you will see that the sub-categories are no longer displayed.

11. Sign in as user **Customer1** (use the My account link at the top right part of the page) and click **Products** in the main menu. You will see a page like the one below. As you can see, the PDAs section and PDA products are not displayed to this user.



Now sign out and sign in again as user **Partner1**.You will all categories and products:



You have learned how to personalize the content displayed to users based on their permissions.

7.10.12 Third-party authentication services

7.10.12.1 Overview

Third-party authentication services provide an alternative way how users can log in and register to your site. This way, they do not need to go through the registration process and create a new username and password for your site. Instead, they can use some username and password that they already use on some popular site (like Windows Live, Facebook, Google, Yahoo!, etc.). Even new users can log on to your site like this, in which case a new user account is created.

Kentico CMS supports the following three authentication providers:

- Windows Live ID
- OpenID
- Facebook Connect

Click the links above to learn more about how to set up authentication using the particular provider.

7.10.12.2 Windows Live ID

7.10.12.2.1 Overview

Windows LiveID is a single sign-on service provided and maintained by Microsoft. By integrating LiveID into your website, you can allow site visitors to log in to your website using their LiveID login and password.

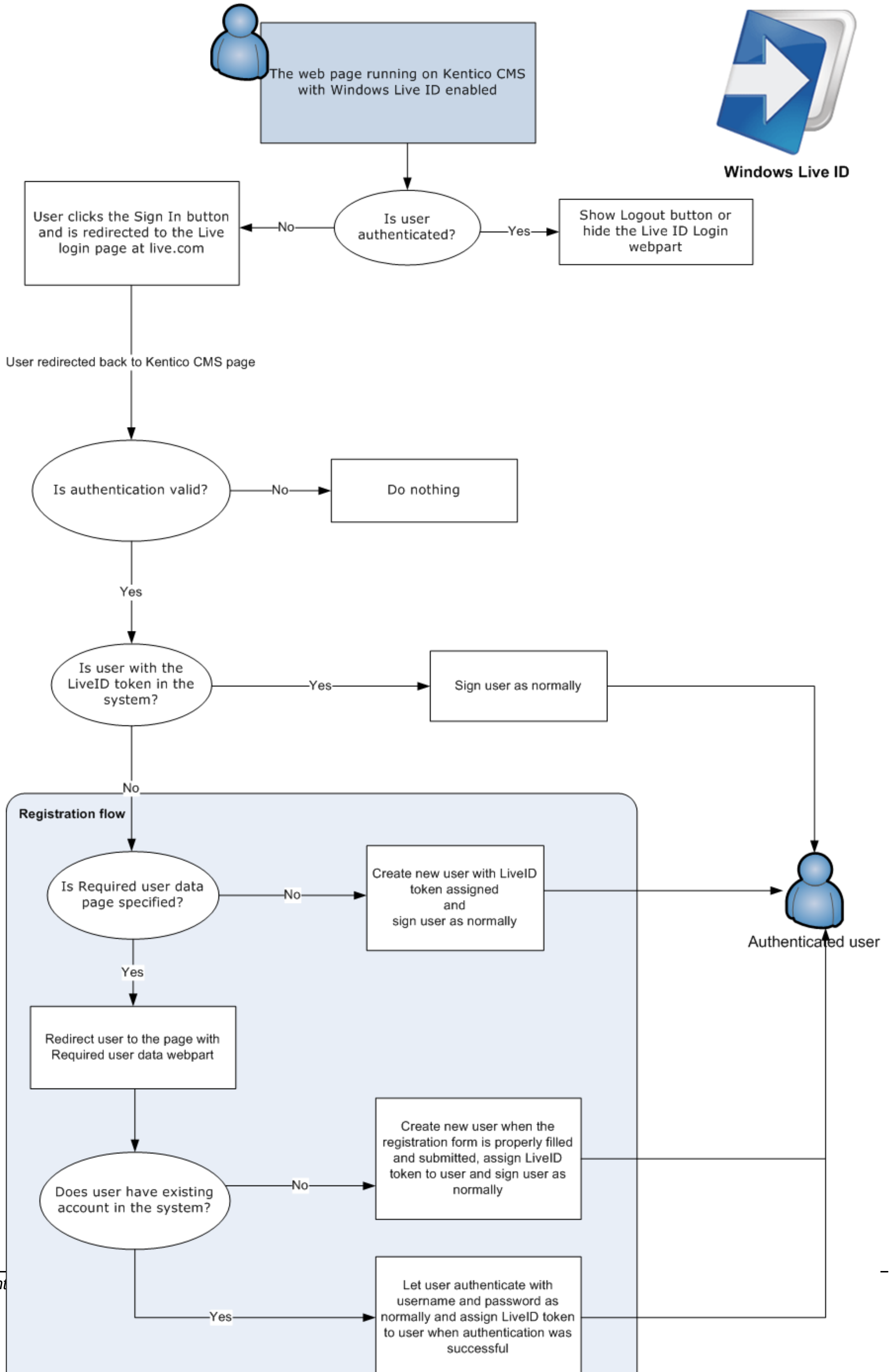
How to start using it

For this to work on your site, you have to do the following things:

1. Register your website at https://manage.dev.live.com/ - to learn how to do it click here
2. Set up Kentico CMS for LiveID authentication - for more details click here
3. Use one of the LiveID web parts on any page of your site - info here

How it works

The following diagram shows how the process of LiveID login works.


The web page running on Kentico CMS with Windows Live ID enabled
Windows Live ID
Is user authenticated?
No
Yes
User clicks the Sign In button and is redirected to the Live login page at live.com
Show Logout button or hide the Live ID Login webpart
User redirected back to Kentico CMS page
Is authentication valid?
No
Do nothing
Yes
Is user with the LiveID token in the system?
Yes
Sign user as normally
No
Registration flow
Is Required user data page specified?
No
Create new user with LiveID token assigned and sign user as normally
Authenticated user
Yes
Redirect user to the page with Required user data webpart
Does user have existing account in the system?
No
Create new user when the registration form is properly filled and submitted, assign LiveID token to user and sign user as normally
Yes
Let user authenticate with username and password as normally and assign LiveID token to user when authentication was successful

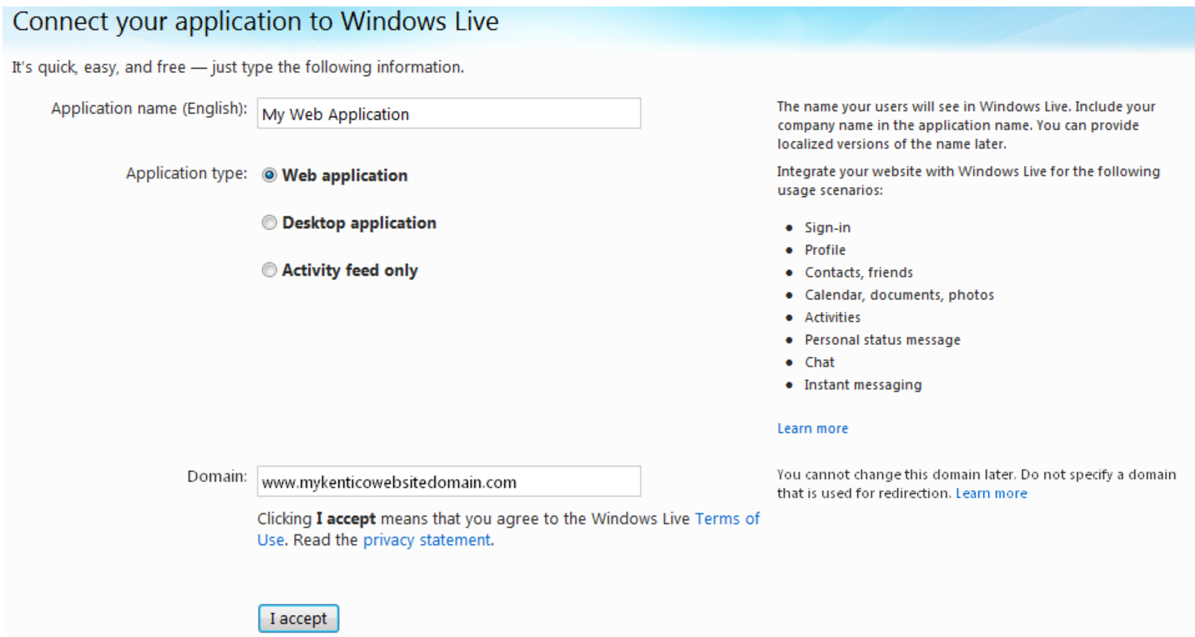
7.10.12.2.2 Registering your application

To enable LiveID logon for your website, you must register it on the Windows Live application management site at the following address: https://manage.dev.live.com/.

1. Once on the page, sign in with your Windows Live ID (you may need to create an account if you don't already have one) and you will be redirected to the application registration page.

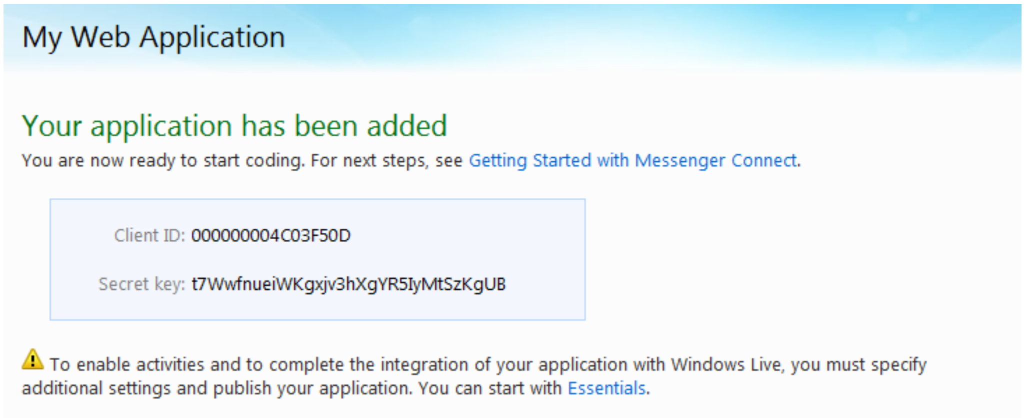
2. Now fill in the following information:

- **Application name** - enter a name that will be used to identify your web application. This name will be displayed in the Windows Live user interface.
- **Application type** - select *Web application.*
- **Domain** - enter the fully qualified domain name of your website.



Read the terms of use and the privacy statement, then click the **I accept** button to register the application.

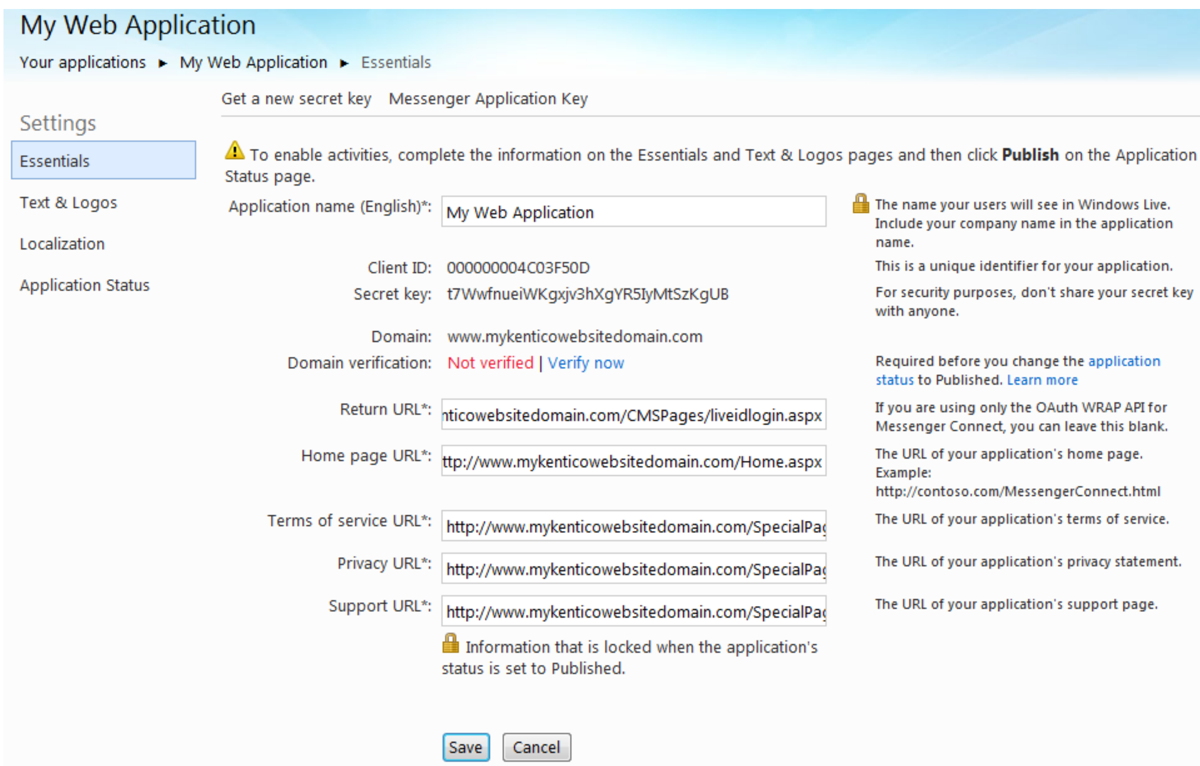
3. You will be shown a screen with important information about the newly registered application. Please make a note of the **Client ID** and **Secret key** values, as they will be required later when filling in Kentico CMS settings.



Next, click on the **Essentials** link in the message below the information box.

4. Fill in the additional settings of the web application:

- **Return URL** - enter a URL in the following format: *http://<yourdomain>/CMSPages/liveidlogin.aspx*, where <*yourdomain*> should be replaced with the domain name of your website (as entered in the *Domain* field).
- **Home page URL**, **Terms of service URL**, **Privacy URL**, **Support URL** - fill in these fields according to the specific structure of your website.



Click **Save**.

5. Now you must verify your ownership of the given domain. Click the **Verify now** link under the **Domain**

field and follow the displayed instructions.

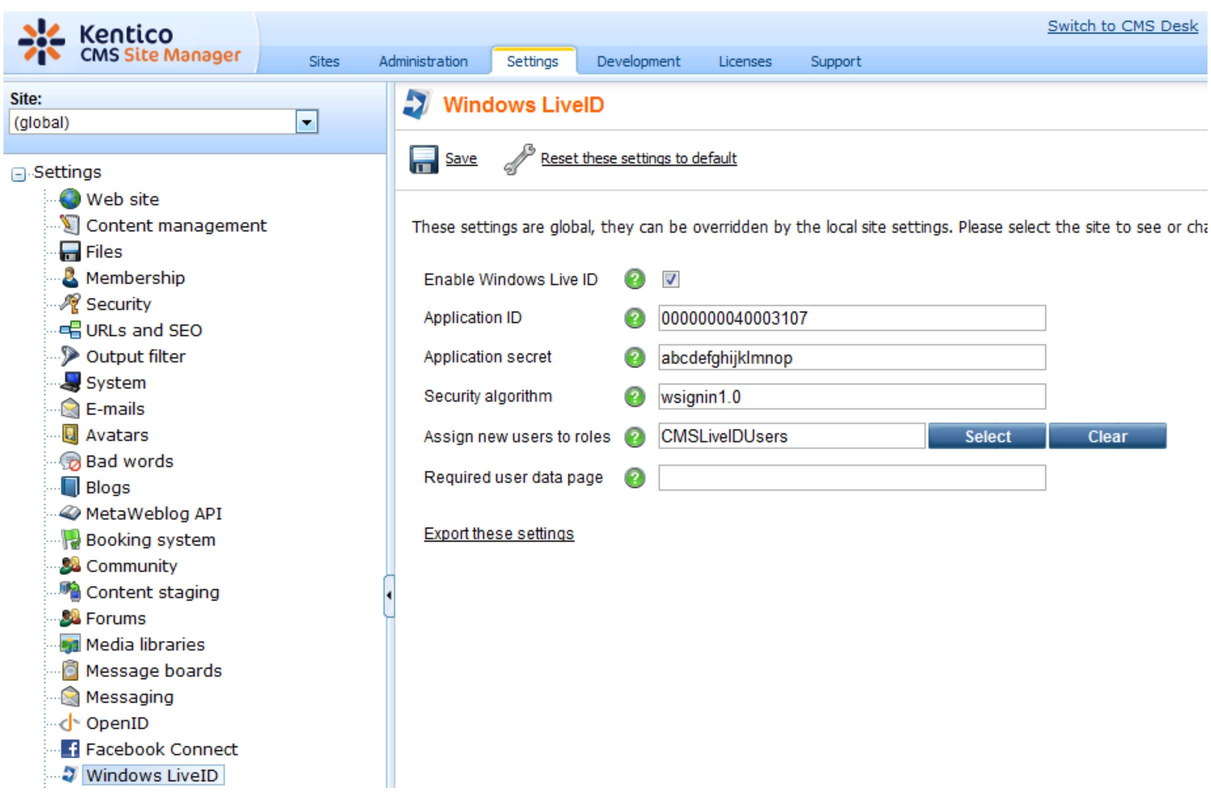
6. Switch to the **Text & Logos** tab, enter a description for the application and upload the logo.

7. Switch to **Application Status** and **Publish** the application.
w

7.10.12.2.3 Settings

LiveID authentication settings are located in **Site Manager -> Settings -> Windows LiveID**. Before you start making the settings, make sure you have the right site selected using the **Site** drop-down list in the top left part of the page.

- **Enable Windows Live ID** - indicates if Windows Live ID authentication should be enabled.
- **Application ID** - identifier of your website. Enter the *Client ID* that you received when registering your website.
- **Application secret** - secret code that will be used for encryption of messages transferred between your website and the LiveID server. Enter the *Secret key* that you received when registering your website.
- **Security algorithm** - algorithm used for the encryption of messages transferred between your website and the LiveID server. It is recommended to use *wsignin1.0*.
- **Assign new users to roles** - new users registered via LiveID login will be assigned to the roles specified here.
- **Required user data page** - URL of a page containing the *Required LiveID user data* web part. If entered, then when a new LiveID user logs in to the site, their user account is not created automatically, but they are redirected to this page and required to enter some additional data (or merge with an existing account) using the web part.



7.10.12.2.4 Available web parts

After registering your web at https://msm.live.com/app and making the necessary settings, you can use the following two web parts on your website to allow authentication via Windows LiveID. The web parts are located under the **Membership** category in the web part catalog.

Windows LiveID

This web part can be used to let site visitors sign in to your website using their Live ID. It can be placed on any page of your website. The web part is hidden to authorized users and will be displayed only to unauthorized public site visitors. With default settings, the web part appears as in the following screenshot.

Although the web part works fine with default settings, it has the following properties to fine-tune its behavior:

- **Sign in text -** if entered, link with the entered text will be used instead of the default sign in image
- **Sign out text -** if entered, link with the entered text will be used instead of the default sign out image
- **Show sign out -** if checked, sign out link will be displayed after the user logs in
- **Show as button -** if checked, buttons will be used instead of links
- **Sign in image -** if set, the image will be used as sign in link
- **Sign out image -** if set, the image will be used as sign out link

Live ID required data

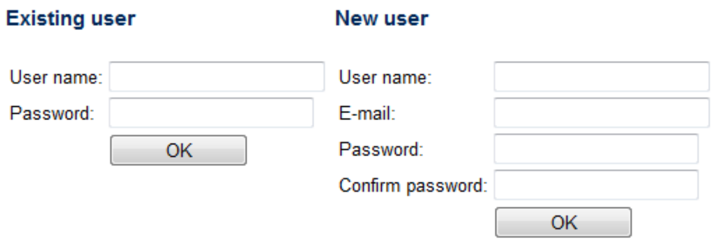
In some cases, you may want new users to provide some extra details before their account is created. For example, if your site is configured to require registration approval or double opt-in, all users need a valid e-mail address to help activate their account.

This can be achieved using the *Live ID user data* web part. The web part must be placed on a page specified by the **Required user data page** in **Site Manager -> Settings -> Windows LiveID**.

The following properties of the web part are the most important:

- **Allow forms authentication -** if checked, new users will be able to enter a password for their new account so that they can log in the usual way as well as via LiveID
- **Allow existing user -** if enabled, users are allowed to join their existing account with their LiveID
- **Default target URL -** if no return URL is passed, users will be redirected to URL entered here after merging or creating the account
- **Hide for no LiveID -** if checked, the web part will be hidden if the page with it is displayed without LiveID logon (e.g. when accessed by entering its URL into the browser)

Other specific web part properties are explained in Kentico CMS Web Parts Reference or after clicking the **Documentation** ( ) link in the top right corner of the web part properties window.



The screenshot above shows how the web part looks with both the **Allow forms authentication** and **Allow existing user** options enabled. In the left part, existing site users can merge their current user account with the LiveID by just entering their user name and password. In this case, the LiveID Token will be added to the LiveID field of the existing user account. New users can enter the required details in the right part of the web part.

7.10.12.3 OpenID

7.10.12.3.1 Overview

OpenID is an open, decentralized standard for authenticating users. It is currently being used by popular websites like Google, Yahoo!, MySpace, Flickr and many more. Logon credentials used on all of these sites can be used to log on to your Kentico CMS site.

How to start using it

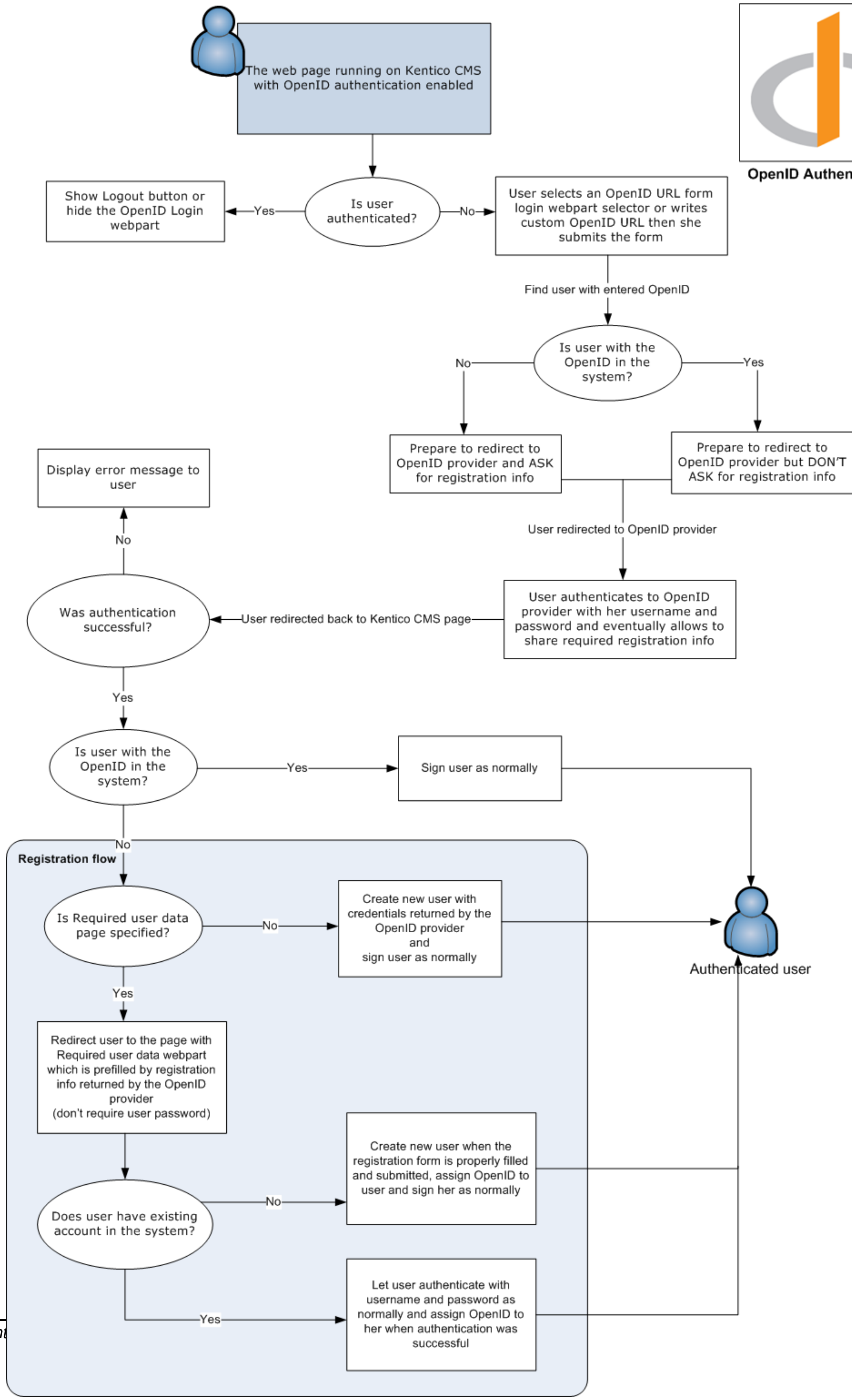
OpenID is the easiest to use of the three supported third-party authentication services. For it to work on your site, you don't need to register your site, all you need to do is take the following two steps:

1. Set up Kentico CMS for OpenID authentication - learn here
2. Use one of the OpenID web parts on any page of your site - info here

How it works

The following diagram shows how the process of OpenID login works:

OpenID Authentication

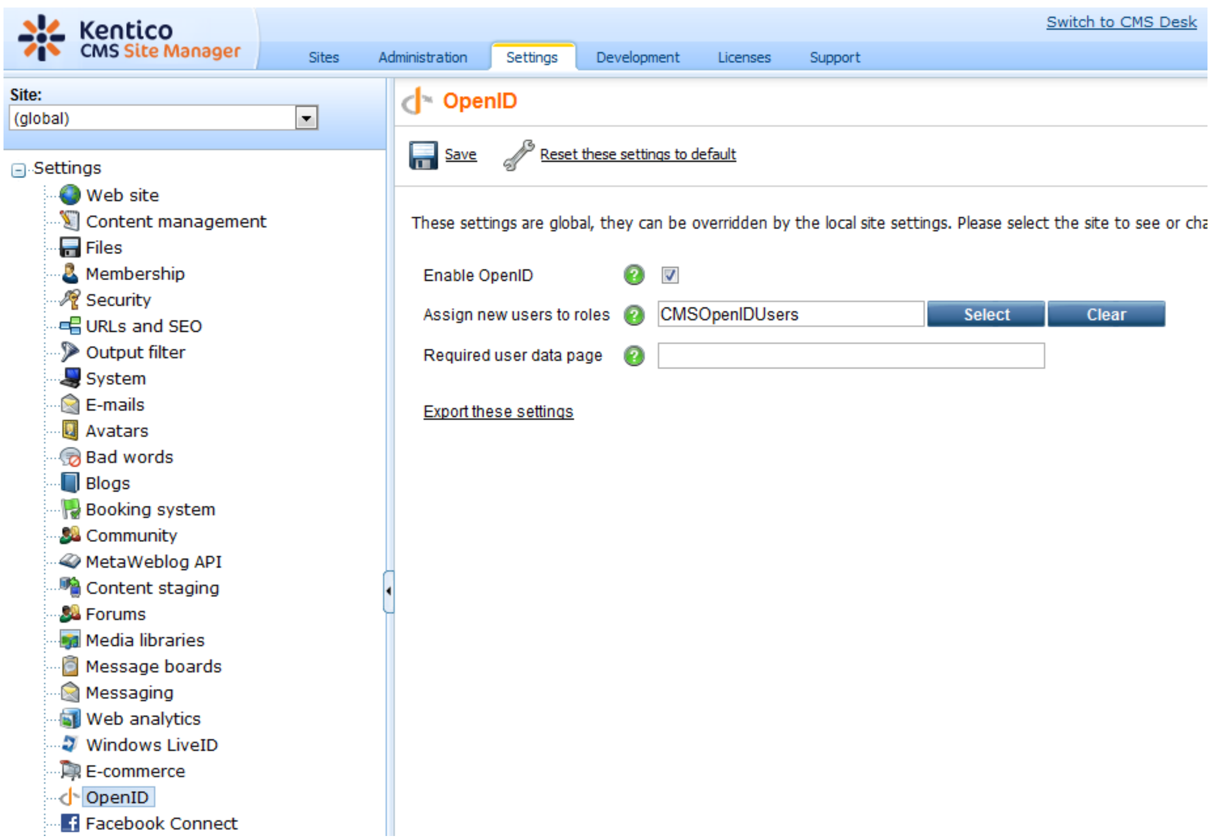


7.10.12.3.2 Settings

First of all, you need to find the **~/Bin/DotNetOpenAuth.dll.rename** file and rename it to **DotNetOpenAuth.dll**. The library is renamed in the default installation because it doesn't support medium-trust environment. Then you need to configure OpenID settings in the administration interface of the CMS.

OpenID authentication settings are located in **Site Manager -> Settings -> OpenID**. Before you start making the settings, make sure you have the right site selected using the **Site** drop-down list at the top left part of the page.

- **Enable OpenID** - indicates if OpenID authentication is enabled
- **Assign new users to roles** - new users registered via OpenID will be assigned to these roles
- **Required user data page** - URL of the page where the *OpenID required data* web part resides; if entered, then when a new OpenID user logs in to the site for the first time, their user account is not created automatically, but they are redirected to this page and required to enter some additional data (or merge with an existing account) using the web part



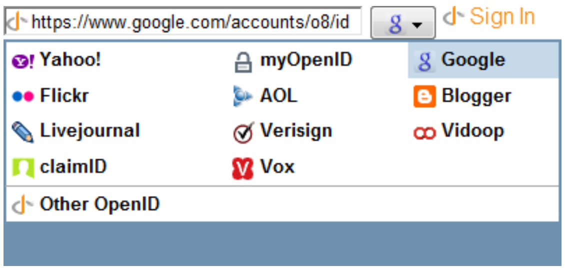
7.10.12.3.3 Available web parts

After configuring the settings, you can use the following two web parts on your website to allow OpenID authentication. The web parts are located under the **Membership** category in the web part catalog.

OpenID logon

This web part can be used to let site visitors sign in to your website using their OpenID. It can be placed on any page of your website. With default settings, the web part appears as in the screenshot below.

It lets you choose from a number of websites which support OpenID and where you might already have an account. If you choose one and click Sign In, you will be redirected to the logon page on that site. After successful logon, you will be redirected back to the Kentico CMS site where you came from, but you will be authenticated and a new account will be created in case this is your first logon.



Even though the web part works fine with default settings, it has the following specific properties to fine-tune its behavior:

- **Providers** - Providers used for OpenID login. Each provider must be specified on a single line. Total number of 3 parameters should be included for each provider:

  1 - provider display name
  2 - provider login URL
  3 - provider icon name placed in *~/CMSWebParts/Membership/OpenID/OpenID_files/*.

  Each parameter must be separated by '|'. The third parameter is optional and if not supplied then the default OpenID icon will be displayed. Provided URL must be the login URL of the OpenID provider. If the username (or blog name, user id, etc.) is part of the URL, then use the *##username##* macro to replace the username part of the URL.

  Example: *myOpenID|http://##username##.myopenid.com/|myopenid.ico*

- **Display textbox** - indicates if the OpenID provider textbox should be visible; if disabled then only the sign in button will be visible
- **Sign in text** - if entered, a link with the entered text will be used instead of the default sign in image
- **Sign out text** - if entered, a link with the entered text will be used instead of the default sign out image
- **Show sign out** - if checked, the sign out link will be displayed after the user logs in
- **Show as button** - if checked, buttons will be used instead of links
- **Sign in image** - if set, the image will be used as the sign in link
- **Sign out image** - if set, the image will be used as the sign out link

- **Required data for new users** - using these settings, you can request additional data from the OpenID provider, which will be added to the newly created user account; in case that you are using the *OpenID required data* web part, the requested data will be pre-filled in the web part; due to the fact that we do not use the latest version of the OpenID library, data may not be obtained successfully from some providers

OpenID required data

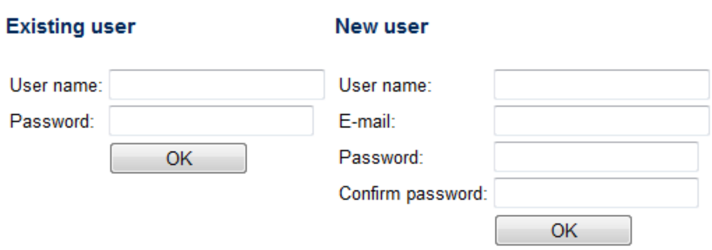
In some cases, you may want new users to provide some extra details before their account is created. For example, if your site is configured to require registration approval or double opt-in, all users need a valid e-mail address to help activate their account.

This can be achieved using the *Open ID user data* web part. The web part must be placed on a page specified by the **Required user data page** value in **Site Manager -> Settings -> OpenID**.

The following properties of the web part are the most important:

- **Allow forms authentication** - if checked, new users will be able to enter a password for their new account so that they can log in the usual way as well as via OpenID
- **Allow existing user** - if enabled, users are allowed to join their existing account with OpenID
- **Default target URL** - if no return URL is passed, users will be redirected to the URL entered here after merging or creating the account
- **Hide for no OpenID** - if checked, the web part will be hidden if the page with it is displayed without OpenID logon (e.g. when accessed by entering its URL into the browser)

Other specific web part properties are explained in Kentico CMS Web Parts Reference or after clicking the **Documentation** ( ) link in the top right corner of the web part properties window.



The screenshot above shows how the web part looks with both the **Allow forms authentication** and **Allow existing user** options enabled. In the left part, an existing site users can merge their current user account with the OpenID by just entering their user name and password. In this case, the OpenID will be added to the OpenID field of the existing user account. New users can enter the required details in the right part of the web part.

7.10.12.4 Facebook Connect

7.10.12.4.1 Overview

Facebook is one of the most popular social networking sites in the world. With its continuously growing number of registered users, it is quite likely that visitors who come to your site will already have a Facebook account. In this case, it is much more convenient for them to use their Facebook logon details than to go through yet another registration on your site. That's why you should consider using Facebook Connect authentication.

How to start using it

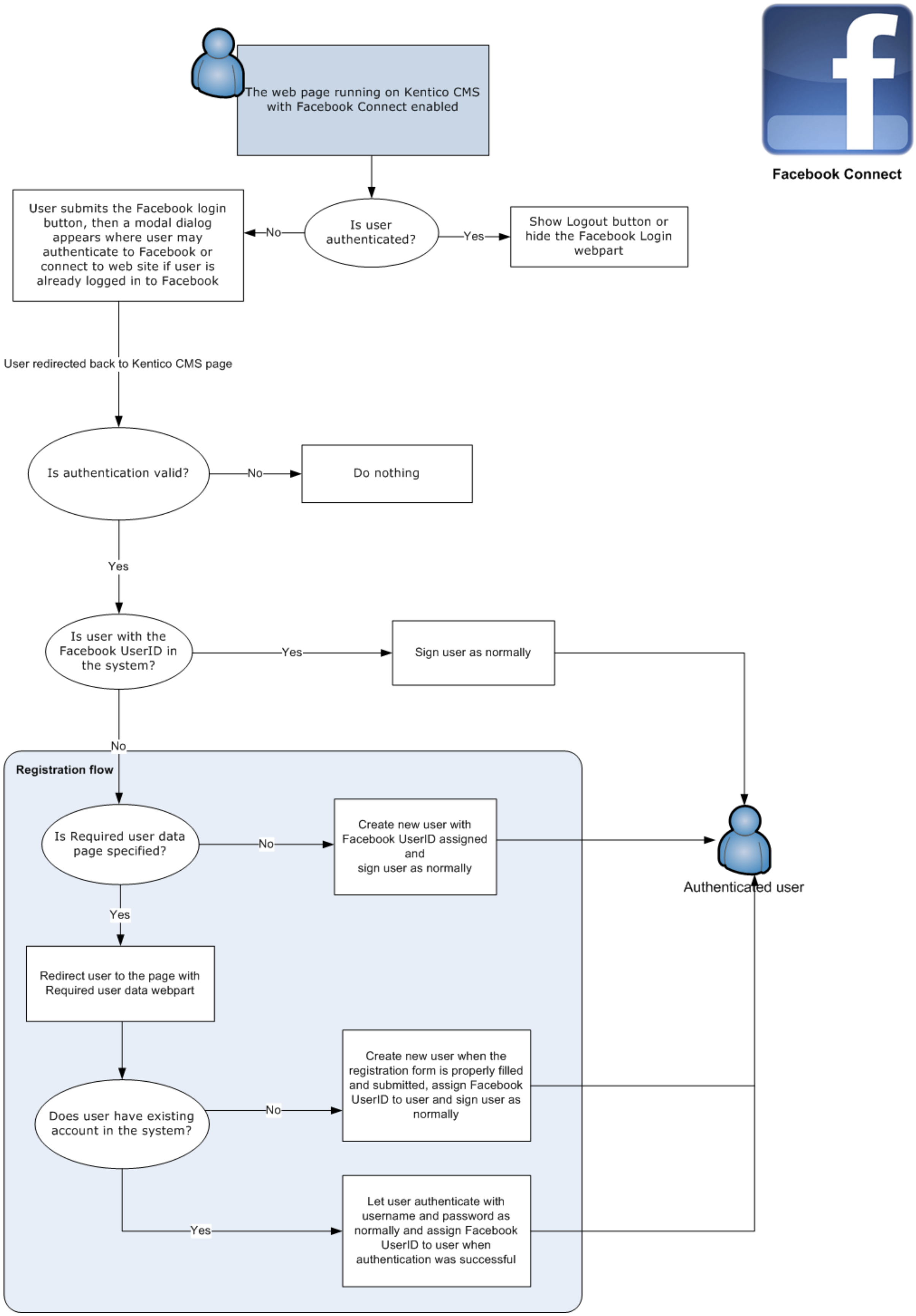
For Facebook Connect to work on your site, you need to register the site, all you need to do is take the following two steps:

1. Register your website at http://www.facebook.com/developers/createapp.php - for more info ready

here
2. Set up Kentico CMS for Facebook Connect authentication - learn here
3. Use one of the Facebook Connect web parts on any page of your site - info here

How it works

The following diagram shows how the process of Facebook Connect login works:


Facebook Connect
The web page running on Kentico CMS with Facebook Connect enabled
Is user authenticated?
No
Yes
User submits the Facebook login button, then a modal dialog appears where user may authenticate to Facebook or connect to web site if user is already logged in to Facebook
Show Logout button or hide the Facebook Login webpart
User redirected back to Kentico CMS page
Is authentication valid?
No
Do nothing
Yes
Is user with the Facebook UserID in the system?
Yes
Sign user as normally
No
Registration flow
Is Required user data page specified?
No
Create new user with Facebook UserID assigned and sign user as normally
Authenticated user
Yes
Redirect user to the page with Required user data webpart
Does user have existing account in the system?
No
Create new user when the registration form is properly filled and submitted, assign Facebook UserID to user and sign user as normally
Yes
Let user authenticate with username and password as normally and assign Facebook UserID to user when authentication was successful

7.10.12.4.2 Registering your application

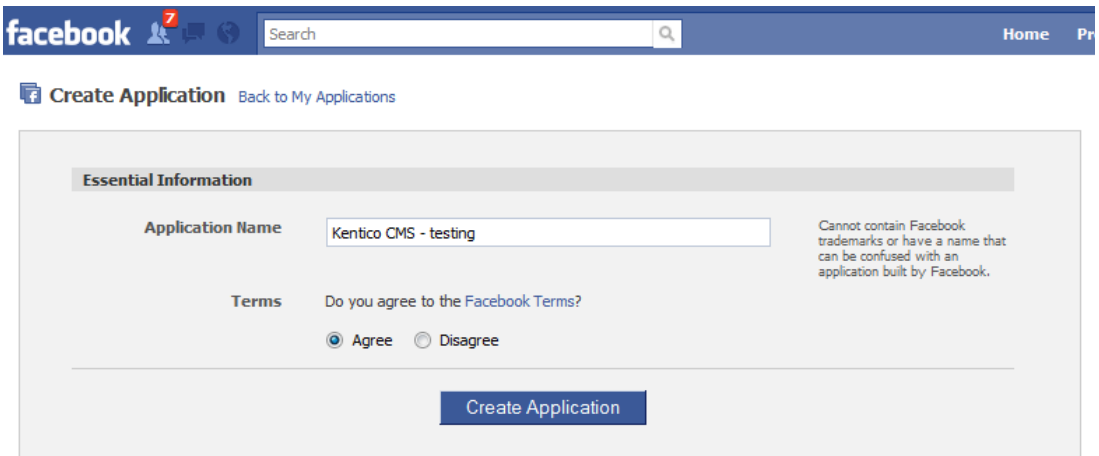
To use Facebook Conntect authentication on your website, you need to register the site at http://www.facebook.com/developers/createapp.php. You need to have a Facebook account to get to the page, so if you don't have one yet, please create it here (you would be prompted to do so otherwise).

The following text describes the registration procedure and points out important information that you obtain during it. Please note that the design of the screenshots below may not be identical as Facebook changes its design quite often.

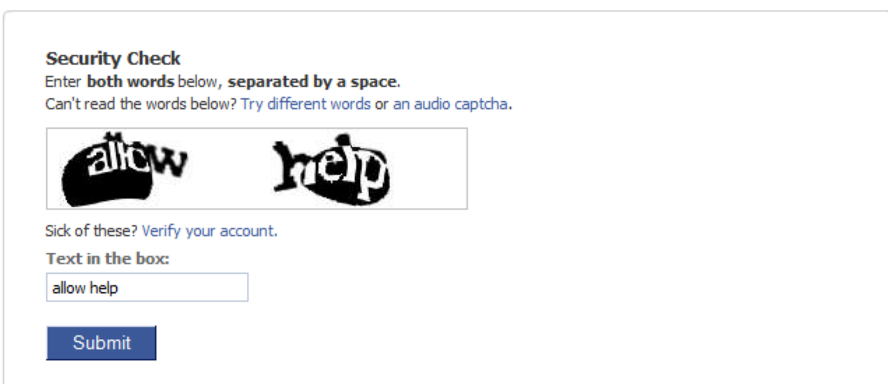
1. With a facebook account created and logged in, visit http://www.facebook.com/developers/createapp.php. Enter the following details into the form:

- **Application Name**: enter the name of your website, e.g. Kentico CMS - testing

Read and **Agree** to the *Facebook Terms* and click **Create Application**.



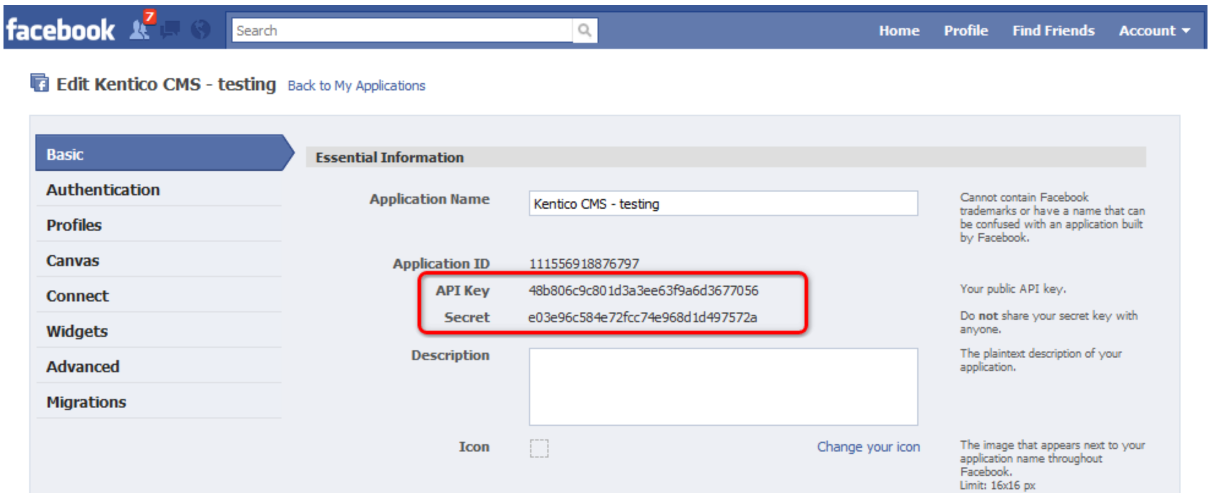
2. Retype the CAPTCHA on and click **Submit**.



3. Your application will be registered and you will see the **Basic** tab of its editing interface. Here, you should focus on the two keys highlighted in the screenshot below, you will need to enter them in **Site Manager -> Settings -> Facebook Connect**:

- **API Key** - this is what you will enter into the **API Key** field
- **Secret** - this is what you will enter into the **Application secret** field

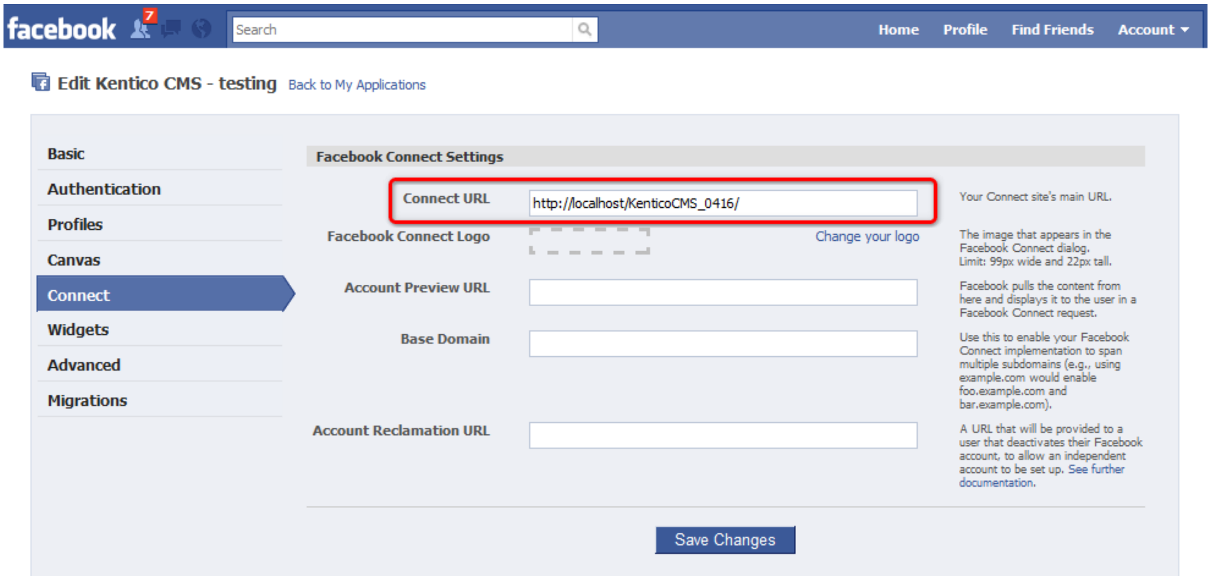
Note them down and set them accordingly in your Kentico CMS instance as described in the Settings chapter.



4. Now switch to the **Connect** tab. Here, you need to enter the URL of your website into the **Connect URL** field. Click **Save Changes**.

Please note!

Using Facebook Connect authentication on **localhost** may result in incorrect behavior of the authentication. This behavior is by design of the Facebook Connect API. To achieve the required functionality, a standard domain name must be used and the domain name entered in this step must match the domain where your web application is really running.

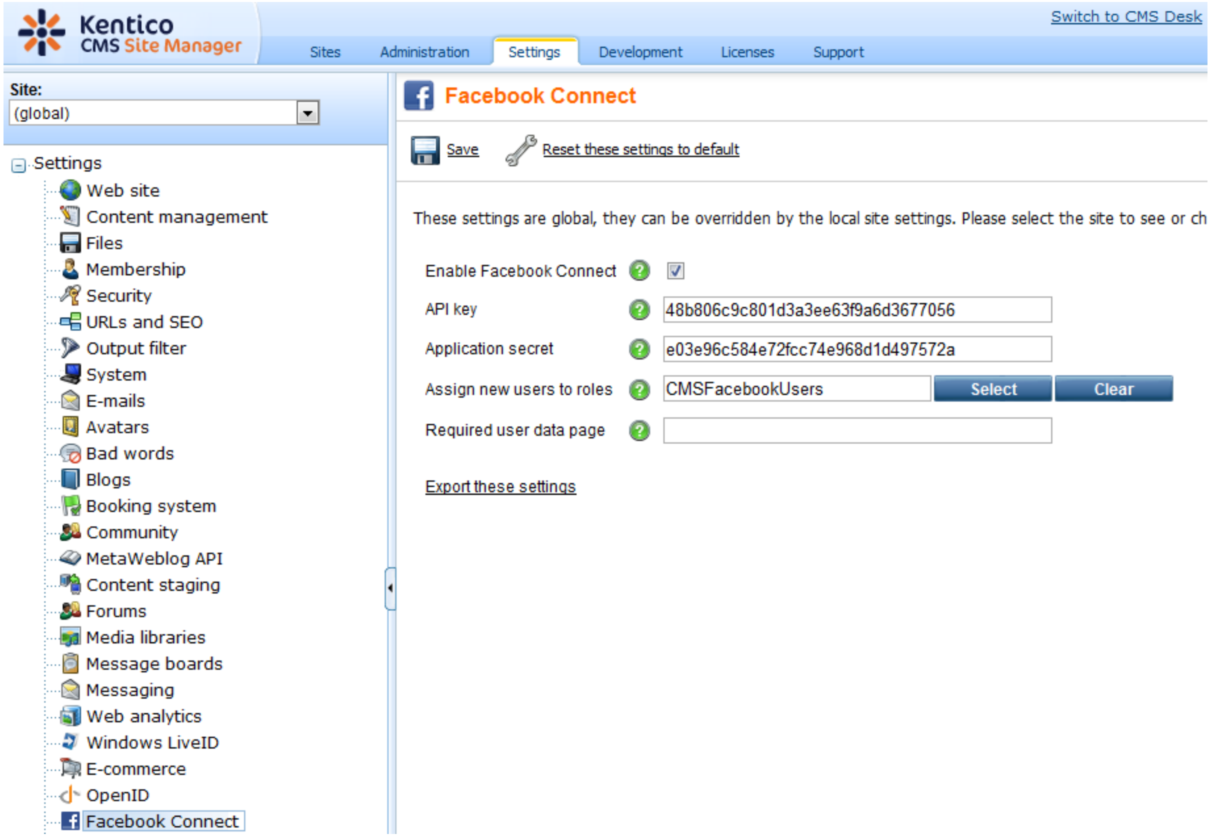


Your web application is now registered at Facebook. With proper settings and web parts, users should now be able to authenticate to your site using their Facebook logon details.

7.10.12.4.3 Settings

Facebook Connect authentication settings are located in **Site Manager -> Settings -> Facebook Connect**. Before you start making the settings, make sure you have the right site selected using the **Site** drop-down list at the top left part of the page.

- **Enable Facebook Connect** - indicates if Facebook Connect authentication is enabled
- **API key** - key obtained during registration of your application on Facebook (as described here)
- **Application secret** - key obtained during registration of your application on Facebook
- **Assign new users to roles** - new users registered via Facebook Connect will be assigned to these roles
- **Required user data page** - URL of the page where the *Facebook Connect required data* web part resides; if entered, then when a new Facebook Connect user logs in to the site, their user account is not created automatically, but they are redirected to this page and required to enter some additional data (or merge with an existing account) using the web part



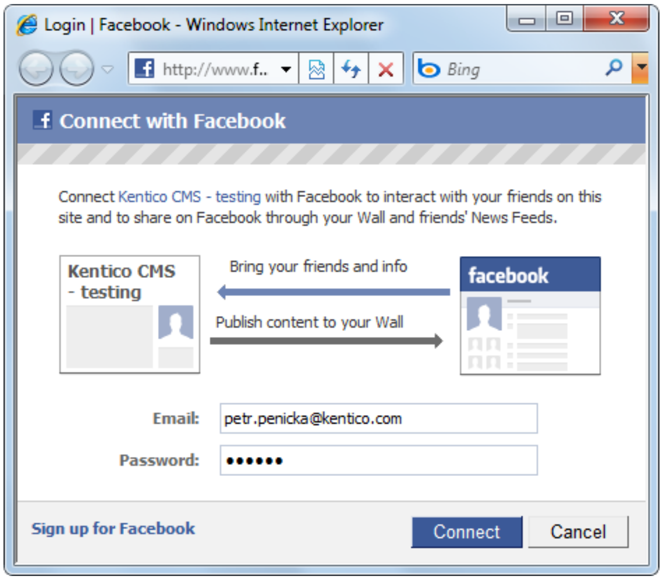
7.10.12.4.4 Available w eb parts

After configuring the settings, you can use the following two web parts on your website to allow Facebook Connect authentication. The web parts are located under the **Membership** category in the web part catalog.

Facebook Connect logon

This web part can be used to let site visitors sign in to your website using their Facebook username and password. It can be placed on any page of your website. The web part is hidden to authorized users and will be displayed only to unauthorized public site visitors. With default settings, the web part appears as in the screenshot below.

If you click the button, a new pop-up window opens, letting you log on using your Facebook username and password.



After successful logon, you will be redirected back to the Kentico CMS site where you came from, but you will be authenticated and a new account will be created in case that this is your first logon.

Even though the web part works fine with default settings, it has the following specific properties to fine-tune its behavior:

- **Sign in text** - if entered, link with the entered text will be used instead of the default sign in image
- **Size** - you can choose from 5 pre-defined sizes of the button
- **Length** - specifies which text label to use on a button with size specified as *small*, *medium*, *large*, or *xlarge*; specify short for the text label *Connect* only or long for the text label *Connect with Facebook*
- **Use latest version** - check this box if you want to use the latest Facebook Connect login buttons; don't check this box if you need to use the original Facebook Connect login buttons

- **Show sign out** - if checked, sign out link will be displayed after the user logs in
- **Sign out text** - if entered, link with the entered text will be used instead of the default sign out image
- **Show as button** - if checked, buttons will be used instead of links
- **Sign out image** - if set, the image will be used as sign out link

Facebook Connect required data

In some cases, you may want new users to provide some extra details before their account is created. For example, if your site is configured to require registration approval or double opt-in, all users need a valid e-mail address to help activate their account.

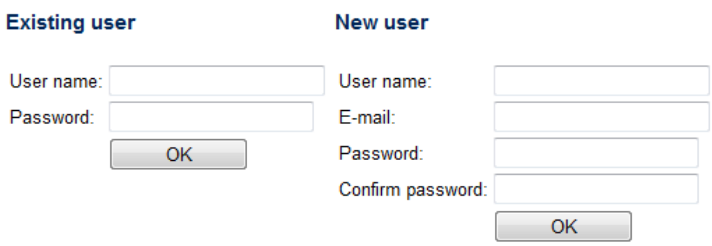
This can be achieved using the *Facebook Connect user data* web part. The web part must be placed on a page specified by the **Required user data page** in **Site Manager -> Settings -> Facebook Connect**.

The following properties of the web part are the most important:

- **Allow forms authentication** - if checked, new users will be able to enter a password for their new account so that they can log in the usual way as well as via Facebook Connect

- **Allow existing user** - if enabled, users are allowed to join their existing account with their Facebook user ID
- **Default target URL** - if no return URL is passed, users will be redirected to the URL entered here after merging or creating the account
- **Hide for no Facebook user ID** - if checked, the web part will be hidden if the page with it is displayed without Facebook Connect logon (e.g. when accessed by entering its URL into the browser)

Other specific web part properties are explained in Kentico CMS Web Parts Reference or after clicking the **Documentation** ( ) link in the top right corner of the web part properties window.



The screenshot above shows how the web part looks with both the **Allow forms authentication** and **Allow existing user** options enabled. In the left part, an existing site user can merge their current user account with the Facebook user ID by just entering their user name and password used on your site. In this case, the Facebook user ID will be added to the **Facebook user ID** field of the existing user account. New users can enter the required details in the right part of the web part.

7.10.12.5 Managing imported users

When a user signs in through a third-party authentication service for the first time, Kentico CMS automatically creates a new user account for them. If you edit such an account in **Site Manager -> Administration -> Users**, you can see that it has the following specific settings:

General tab

On the General tab, you can notice that the **User name** and **Full name** fields have been automatically filled with a string in the following format:

User name

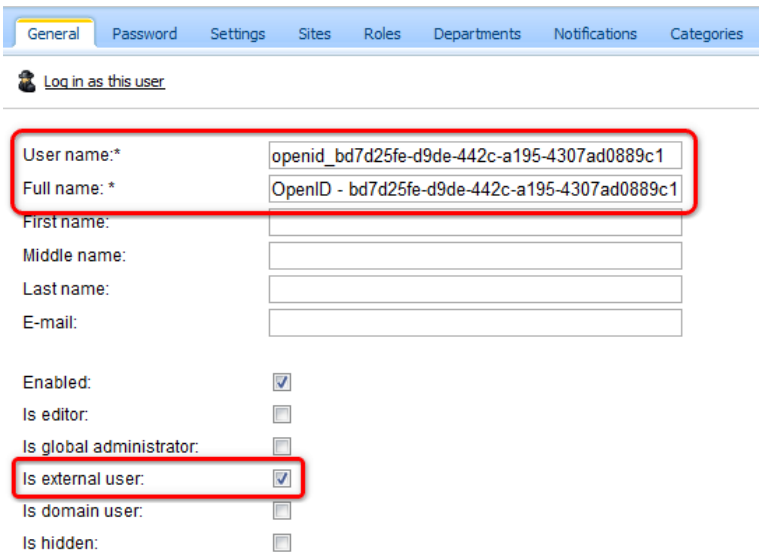
- **Live ID**: *liveid_<liveidtoken>*
- **OpenID**: *openid_<openid>*
- **Facebook Connect**: *facebookid_<facebookuserid>*

Full name

- **Live ID**: *LiveID - <liveidtoken>*
- **OpenID**: *OpenID - <openid>*
- **Facebook Connect**: *Facebook ID - <facebookuserid>*

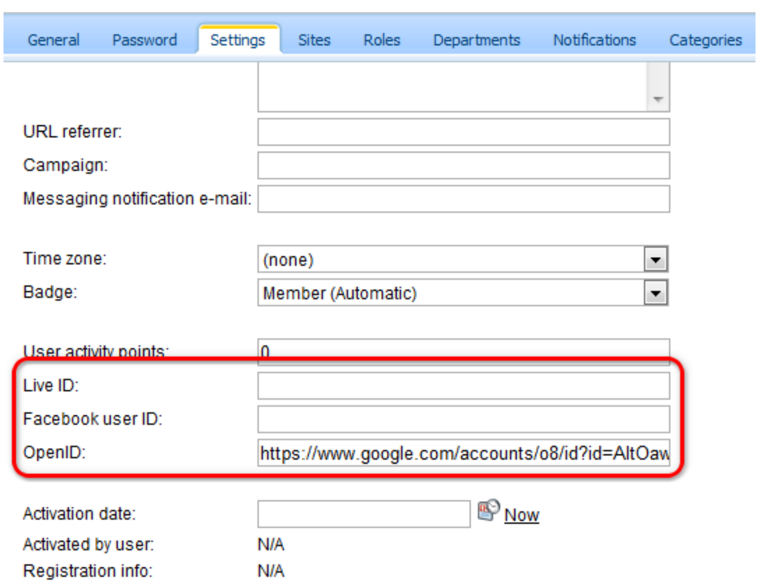
These values can be changed manually without any effects on the functionality.

You can also notice that the **Is external user** check-box is enabled. It indicates that the user account is imported from some external user database and disables forms authentication for the user, i.e. the user can only log in using the third-party authentication service. When merging an existing account with third-party authentication using one of the **<provider> required user data** web parts, the existing account that you want to merge must not have this option enabled (because forms authentication using the web part would not work).



If you switch to the **Settings** tab, you can notice the **Live ID**, **Facebook user ID** and **OpenID** fields. This is where the user's ID from the particular provider is stored.

You can change the values manually if you need to. You just need to make sure that the entered ID is valid. In such a case, the newly entered ID will be used when the user logs in. You can also delete the value, in which case no ID will be assigned to the user. In this case, please remember to disable the **Is external user** option on the **General** tab so that the user can log in using forms authentication.



7.10.13 Membership internals and API

7.10.13.1 Database tables and API classes

The the following database tables are used to store membership data:

- **CMS_User** - contains users and their basic properties
- **CMS_UserSettings** - contains user settings
- **CMS_Role** - contains roles and information about them
- **CMS_Permission** - contains permissions and information about them
- **CMS_Badge** - contains badges and their properties

The Membership API is provided by the following **CMS.SiteProvider** namespace classes:

- **UserInfo**, **UserInfoProvider** - these classes provide functionality for managing users
- **UserSettingsInfo**, **UserSettingsInfoProvider** - these classes provide functionality for managing user settings
- **UserSiteInfo**, **UserSiteInfoProvider** - these classes allow the assigning of users to sites
- **RoleInfo**, **RoleInfoProvider** - these classes provide functionality for managing roles
- **PermissionNameInfo**, **PermissionNameInfoProvider** - these classes provide functionality for managing permissions
- **RolePermissionInfo**, **RolePermissionInfoProvider** - these classes allow the assigning of permissions to roles
- **BadgeInfo**, **BadgeInfoProvider** - these classes provide functionality for badges

The following topics show examples of how these classes can be used:

- Managing users
- Managing roles and permissions
- Managing badges

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

7.10.13.2 Managing users

The following sample code shows how you can create a new user:

[C#]

```
using CMS.SiteProvider;

...

        // Create new user object
        UserInfo ui = new UserInfo();

        // Set properties
```

```
        ui.UserName = "Username";
        ui.FirstName = "John";
        ui.LastName = "Jones";
        ui.FullName = ui.FirstName + " " + ui.LastName;
        ui.UserNickName = "nickname";
        ui.IsEditor = false;
        ui.IsGlobalAdministrator = false;
        ui.Email = "john.jones@localhost.local";
        ui.Enabled = true;

        // Save object to database
        UserInfoProvider.SetUserInfo(ui);
```

The following sample code shows how you can add a user to the current site:

[C#]

```
using CMS.SiteProvider;
using CMS.CMSHelper;

...

        // Get user object
        UserInfo ui = UserInfoProvider.GetUserInfo("Username");

        // If user exists
        if (ui != null)
        {
            // Add user to site
            UserSiteInfoProvider.AddUserToSite(ui, CMSContext.CurrentSite);
        }
```

The following sample code shows how you can assign an avatar to a user:

[C#]

```
using CMS.SiteProvider;

...

        // Get user object
        UserInfo ui = UserInfoProvider.GetUserInfo("Username");

        // Get avatar object
        AvatarInfo ai = AvatarInfoProvider.GetAvatarInfoWithoutBinary("custom");

        // If both exist
        if ((ui != null) && (ai != null))
        {
            // Add avatar to user settings
```

```
            ui.UserSettings.UserAvatarID = ai.AvatarID;

            // Update user object in database
            UserInfoProvider.SetUserInfo(ui);

        }
```

7.10.13.3 Managing roles and permissions

The following sample code shows how you can create a new role and add it to the current site:

[C#]

```
using CMS.SiteProvider;
using CMS.CMSHelper;

...

        // Create empty role object
        RoleInfo ri = new RoleInfo();

        // Set properties
        ri.DisplayName = "Custom role";
        ri.RoleName = "CustomRole";
        ri.SiteID = CMSContext.CurrentSiteID;

        // Save role object to database
        RoleInfoProvider.SetRoleInfo(ri);
```

The following sample code shows how you can add a user to a role:

[C#]

```
using CMS.SiteProvider;
using CMS.CMSHelper;

...

        // Get role object
        RoleInfo ri = RoleInfoProvider.GetRoleInfo("CustomRole", CMSContext.CurrentS

        // Get user object
        UserInfo ui = UserInfoProvider.GetUserInfo("Username");

        // If both exist
        if ((ri != null) && (ui != null))
        {
            // Create new user role object
            UserRoleInfo uri = new UserRoleInfo();
```

```
            // Set properties
            uri.UserID = ui.UserID;
            uri.RoleID = ri.RoleID;

            // Save to database
            UserRoleInfoProvider.SetUserRoleInfo(uri);
        }
```

The following sample code shows how you can add a permission to a role:

[C#]

```
using CMS.SiteProvider;
using CMS.CMSHelper;

...

        // Get role object
        RoleInfo ri = RoleInfoProvider.GetRoleInfo("CustomRole",CMSContext.CurrentSi

        // Get permission object
        PermissionNameInfo pni = PermissionNameInfoProvider.GetPermissionNameInfo("R

        // If both exist
        if ((ri != null) && (pni != null))
        {
            // Create new role permission object
            RolePermissionInfo rpi = new RolePermissionInfo();

            // Set properties
            rpi.RoleID = ri.RoleID;
            rpi.PermissionID = pni.PermissionId;

            // Save to database
            RolePermissionInfoProvider.SetRolePermissionInfo(rpi);
        }
```

7.10.13.4 Managing badges

The following sample code shows how you can get a badge as a *BadgeInfo* object:

[C#]

```
using CMS.SiteProvider;

...

        // Get Badge object by ID
        BadgeInfo bi = BadgeInfoProvider.GetBadgeInfo(1);
```

```
        // Get Badge object by badge code name
        BadgeInfo bi2 = BadgeInfoProvider.GetBadgeInfo("Admin");
```

The following sample code shows how you can get a DataSet containing badges from the system:

[C#]

```
using System.Data;
using CMS.SiteProvider;

...

        string where = "BadgeTopLimit > 50";
        string orderby = "BadgeTopLimit DESC";
        int topN = 10;
        string selectedColumns = "BadgeID, BadgeTopLimit, BadgeImageURL";

        // Get dataset of badges according to the given parameters
        DataSet ds = BadgeInfoProvider.GetBadges(where, orderby);

        DataSet ds2 = BadgeInfoProvider.GetBadges(where, orderby, topN, selectedColu
```

The following sample code shows how you can create a new badge:

[C#]

```
using CMS.SiteProvider;

...

        // Create new badge object
        BadgeInfo bi = new BadgeInfo();

        // Set properties
        bi.BadgeDisplayName = "Test badge";
        bi.BadgeName = "TestBadge";
        bi.BadgeTopLimit = 50;
        bi.BadgeImageURL = "~/App_Themes/Default/Images/Objects/CMS_Badge/Default/
siteadmin.gif";
        bi.BadgeIsAutomatic = true;

        // Save to database
        BadgeInfoProvider.SetBadgeInfo(bi);
```

The following sample code shows how you can modify the display name of an existing badge:

[C#]

```
using CMS.SiteProvider;

...

        // Get badge object
        BadgeInfo bi = BadgeInfoProvider.GetBadgeInfo("TestBadge");

        // If badge exists
        if (bi != null)
        {
            // Update property
            bi.BadgeDisplayName += " updated";

            // Save changed object to database
            BadgeInfoProvider.SetBadgeInfo(bi);
        }
```

The following sample code shows how you can delete a badge from the system:

[C#]

```
using CMS.SiteProvider;

...

        // Get badge object
        BadgeInfo bi = BadgeInfoProvider.GetBadgeInfo("TestBadge");

        // If badge exists
        if (bi != null)
        {
            // Delete from database
            BadgeInfoProvider.DeleteBadgeInfo(bi);
        }
```

The following sample code shows how you can update the activity points of a user:

[C#]

```
using CMS.SiteProvider;
using CMS.CMSHelper;

...

        // Get user
       UserInfo ui = UserInfoProvider.GetUserInfo("gold");

       // If user exists
       if (ui != null)
       {
```

```
            // Update activity points of user
            BadgeInfoProvider.UpdateActivityPointsToUser(ActivityPointsEnum.
BlogCommentPost, ui.UserID, CMSContext.CurrentSiteName, true);
       }
```

The following sample code shows how you can assign a badge to a user:

[C#]

```
using CMS.SiteProvider;

...

       // Get user object
       UserInfo ui = UserInfoProvider.GetUserInfo("Username");

       // Get badge object
       BadgeInfo bi = BadgeInfoProvider.GetBadgeInfo("Member");

       // If both exists
       if ((ui != null) && (ui != null))
       {
           // Add badge to user settings
           ui.UserSettings.UserBadgeID = bi.BadgeID;

           // Update user object in database
           UserInfoProvider.SetUserInfo(ui);

       }
```

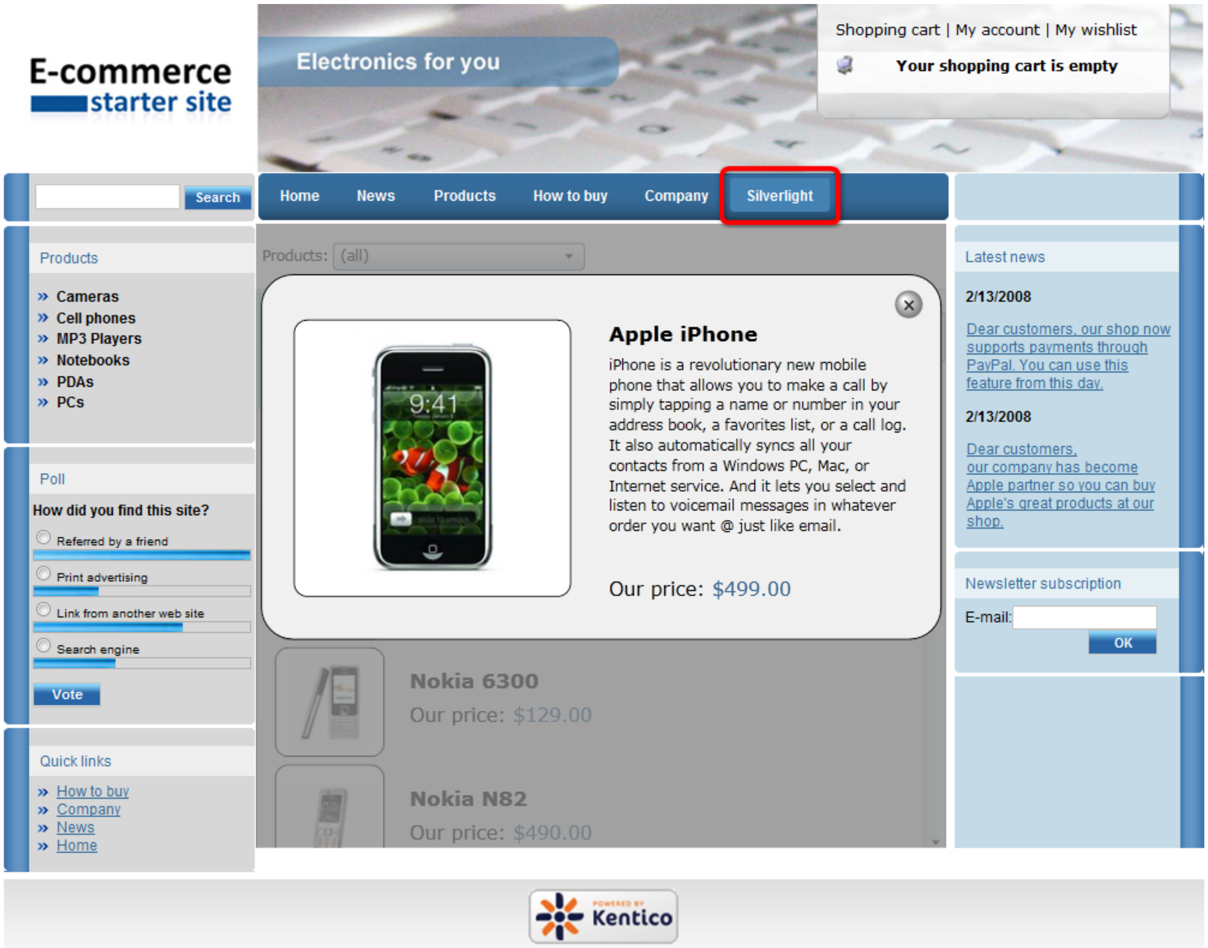
7.11 Microsoft Silverlight integration

7.11.1 Overview

Kentico CMS comes with native support of **Microsoft Silverlight 2.0**. Microsoft Silverlight is a new cross-browser, cross-platform technology for building and delivering the next generation of media experiences and Rich Interactive Applications (RIA) for the Web.

Silverlight applications run in the internet browser. All you need is a small plug-in installed in your browser. The plug-in is free and in case that users access a site containing a Silverlight application without this plug-in installed, an install banner leading to the download link will be offered automatically.

You can find an example of a Silverlight application on the **E-commerce Site** sample website, in the **Silverlight** section.



How it works in general

1. The developer creates a website with a built-in Silverlight application.
2. The site visitor navigates to that site using an internet browser.
3. If the user does not already have the required plug-in installed in the browser, they are automatically prompted to install it.
4. The Silverlight application is executed.

Creating Silverlight applications

Silverlight is a **.NET Framework** based technology, so if you are familiar with development using **Visual Studio** and one of the **.NET Framework languages** like **C#**, it will be much easier for you to learn Silverlight. For developing Silverlight 2.0 applications, you will need at least **Microsoft Visual Studio 2008 SP1** with **Silverlight Tools**. There is one more powerful tool for designers **- Expression Blend**, which enables you to create application design in a really comfortable way. We also strongly recommend installing the **Silverlight Toolkit**, which brings many new controls that can be used in your Silverlight applications.

Visit the Silverlight community site where you can download all the required components. What is more, you can find there valuable tutorials which can help you get started in the development of Silverlight applications.

We also recommend reading the official Microsoft Silverlight documentation.

- To learn how to add a Silverlight application to your site, please refer to the Adding a Silverlight application to your site topic.
- If you would like to learn how to configure Internet Information Services (IIS) to enable Silverlight applications on your site, please refer to the IIS configuration topic.

7.11.2 Adding a Silverlight application to your site

To add a Silverlight application to your site, you will need to use the **Silverlight application** web part, which is a container for Silverlight applications. You can find a live example of use of this web part on the **E-commerce Site** sample website, in the **Silverlight** section.

The following steps need to be taken to add a Silverlight application to a website.

1. Go to **CMS Desk** and from the content tree, choose the page where you would like to add your Silverlight application.

2. Switch to the **Design** tab and click the **Add web part** () icon at the top right-hand corner of the web part zone where you want to place the application.

3. Choose the **Silverlight\Silverlight application** web part and click **OK**.

4. In the web part properties window which pops-up, you can set the following web-part-specific properties:

- **Application path** - path to your Silverlight application; e.g. *~/ClientBin/MyApplication.xap*
- **Minimum version** - minimum version of Silverlight required by the silverlight application to run by this web part
- **Container width** - width of the application container; can be entered either as an integer value (e.g. *315*) or as a percentage value (e.g. *59%*)
- **Container height** - height of the application container; can be entered either as an integer value (e.g. *315*) or as a percentage value (e.g. *59%*)
- **Container background** - background of the application container; can be entered in hexadecimal (e.g. *#323232*) format or selected from a Color picker, which is displayed after clicking the Select button; if no value is entered, white color is used
- **Endpoint address** - web service endpoint address the client application can connect to; if specified, its value is added as parameter with 'endpoint' key to the application parameters collection; you need to handle this parameter in your Silverlight application for it to take effect
- **Parameters** - Silverlight application parameters in the following format: *<key1>=<value1>, <key2>=<value2>,…*
- **Alternate content** - custom HTML content which is displayed to users when Silverlight plug-in is not installed; leave blank if you want the default alternate content to be displayed

Click **OK**, switch to the **Live site** and enjoy your Silverlight application running on your Kentico CMS website.

7.11.3 IIS configuration

Silverlight introduces two new file extensions:

- **.xaml** for XAML files
- **.xap** for the zip-based binary packaging format

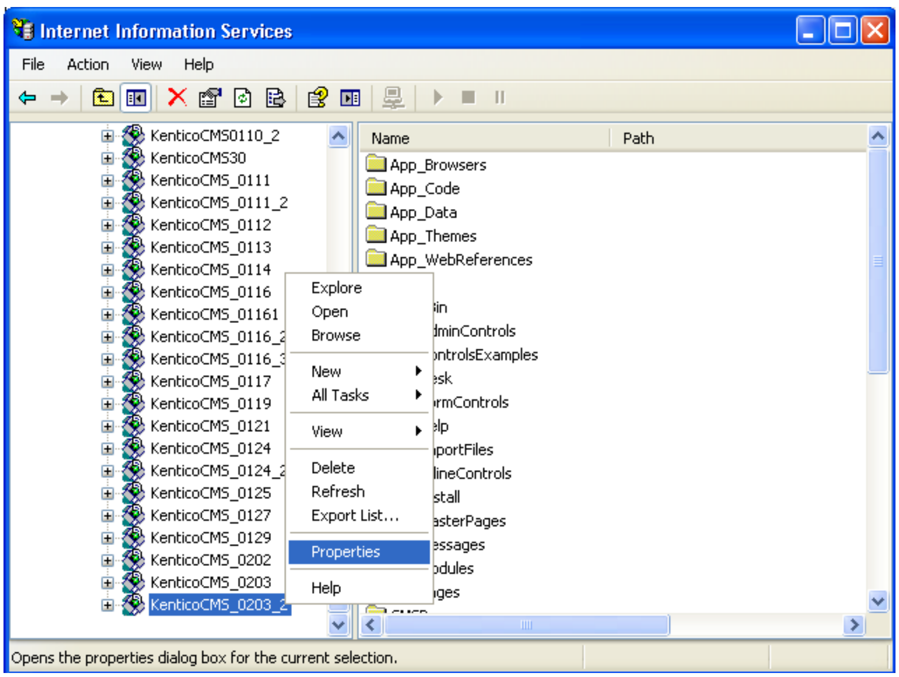
For your application to work correctly, you need to add the MIME types for these file extensions to your web server so that it recognizes Silverlight content appropriately:

- **Extension**: .xaml
- **MIME type**: application/xaml+xml

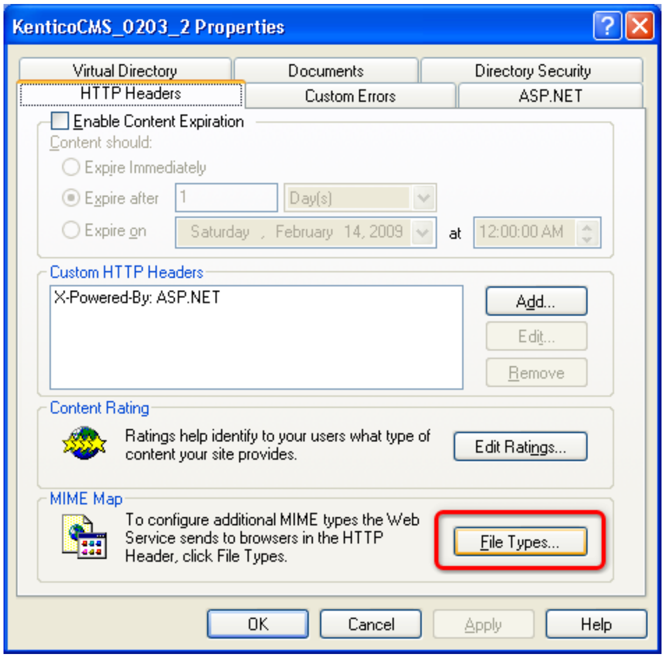
- **Extension**: .xap
- **MIME type**: application/x-silverlight-app

Here is a step-by-step guide on how to do it on IIS 5.1:

1. Open IIS, right-click your website and choose **Properties**.



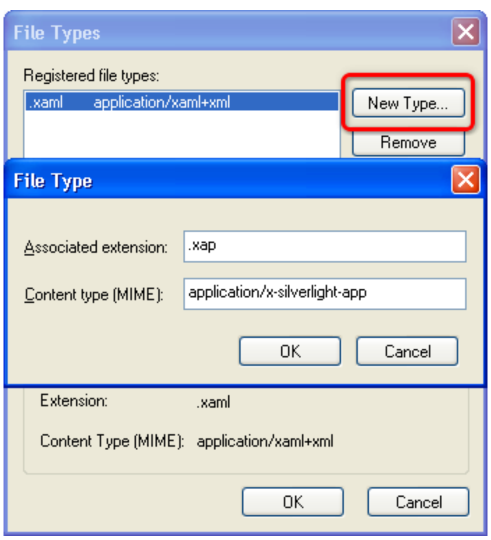
2. Switch to the **HTTP Headers** tab and choose **File Types** in the **MIME Map** section.



3. Using the **New Type** button, add the following MIME types:

- **Associated extension**: .xaml
- **Content type (MIME)**: application/xaml+xml

- **Associated extension**: .xap
- **Content type (MIME)**: application/x-silverlight-app



7.12 Multilingual and international support

7.12.1 Overview

Language settings of Kentico CMS can be performed on two levels: **User interface culture** defines the language of the administration interface (CMS Desk and Site Manager), while **Content culture** is used to define the language of your website content.

User interface culture

The user interface culture represents language of the administration interface - CMS Desk and Site Manager. UI cultures can be managed and added in **Site Manager -> Development -> UI Cultures**.

UI cultures can be set for each user in **Site Manager -> Administration -> Users** -> edit () a user -> **General** -> **Preferred user interface culture**.

The default UI culture can be set by adding the following key to the <*appSettings*> section of your site's *web.config* file:

```
<add key="CMSDefaultUICulture" value=”en-nz” />
```

If you use this key, you also need to rename the **~\CMSResources\CMS.resx** file to **CMS.en-us.resx** and the **CMS.en-nz.resx** file to **CMS.resx**. This is needed because the *CMS.resx* file is used when the *(default)* option in selected in users' *Preferred user interface culture*.

When this key is used and the CMS.resx file contains the en-nz dictionary, for the user who has their **Preferred user interface culture** set to **(default)**, the UI culture will be *en-nz*.

Learn more about configuring multilingual UI here.

Content culture

The content culture determines the actual culture of the website content stored in Kentico documents. In a multilingual site, there are multiple versions of a document for each culture.

The list of all content cultures available in Kentico CMS can be found in **Site Manager -> Development -> Cultures**. All major cultures are provided in the list.

The default content culture can be set the following ways:

- **Site Manager -> Sites -> Edit -> Default content culture**
- **Site Manager -> Settings -> (select site from the drop-down list) -> website -> Default culture of the content**

See the example on this page.

Based on the settings of the user's **Preferred content culture** in the **Site Manager -> Administration -> Users ->**edit () a user -> **General** tab the user will either see the content in the default culture - if the setting is to **(default)** - or in the preferred culture.

It is also possible to set the default culture separately for anonymous visitors of the site. This again can be set on two levels:

- generally for a site in **Site Manager -> Sites ->** edit () a site **-> General -> Default visitor culture**
- for a particular domain alias of a site in **Site Manager -> Sites ->** edit () a site**-> Domain Aliases ->** edit () an alias **-> Default visitor culture**

It is either set to **(Automatic)** in which case the **user's browser settings** will be used, or it can be set manually to one of the available cultures. The list of available cultures can be found in **Site Manager -> Sites ->** edit () a site **-> Cultures**.

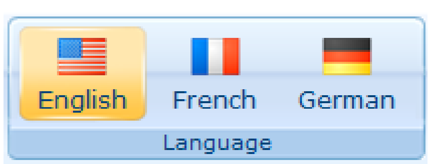
7.12.2 Configuring multilingual content

This chapter explains how to configure website for displaying content in multiple languages. When you create a new website, it uses the culture you specify during the New site wizard.

The following example shows how to add extra language versions to the standard Corporate Site:

1. Go to **Site Manager -> Sites** and click the **Edit site** () icon next to the Corporate Site. Go to the **Cultures** tab. Click the **Add cultures** button and add the following cultures: French - France and German - Germany. Click **Save**.

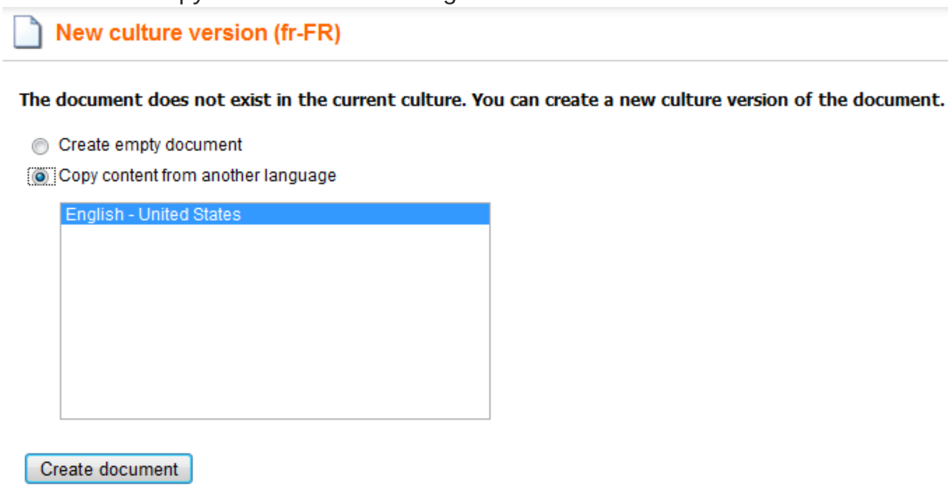
2. Switch to the CMS Desk. You can see there's a new section with language selection in the main toolbar:



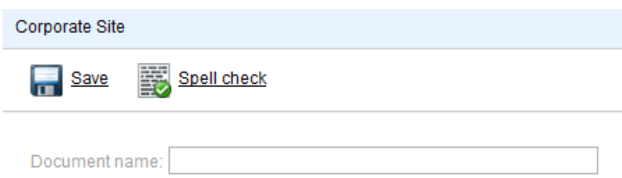
3. Click root document and choose **French** culture. Since the French version of the document doesn't exist yet, you're offered with two options:
- **create empty document** - this option creates a new document of the same type without any content.
- **copy content from another language** - this option creates a new document of the same type and

copies the content from the English version

Choose to copy content from the English version and click **Create document**:

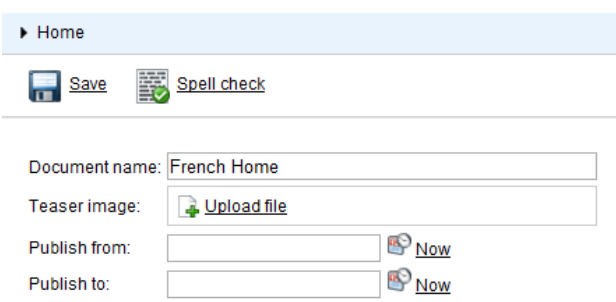


4. You are redirected to the editing form of the new version:



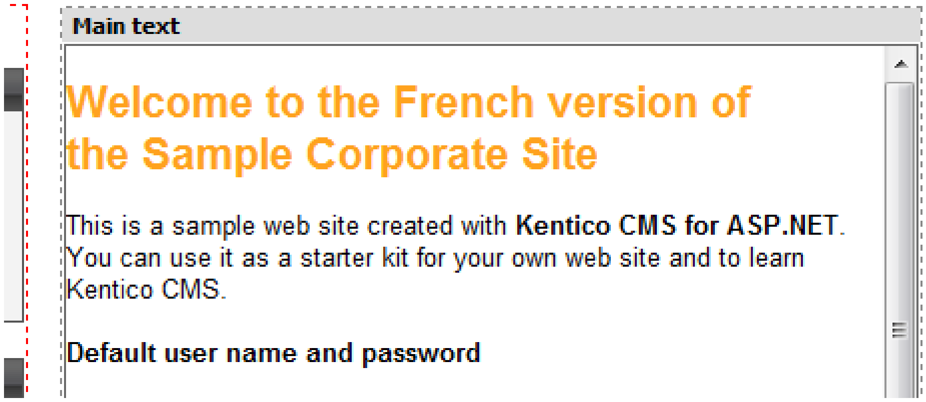
The root cannot have any name, so the box is now disabled. Click **Save**. The new document culture version is created.

5. Now repeat the same for the **Home** page. On the editing form, change the document name to: French Home.



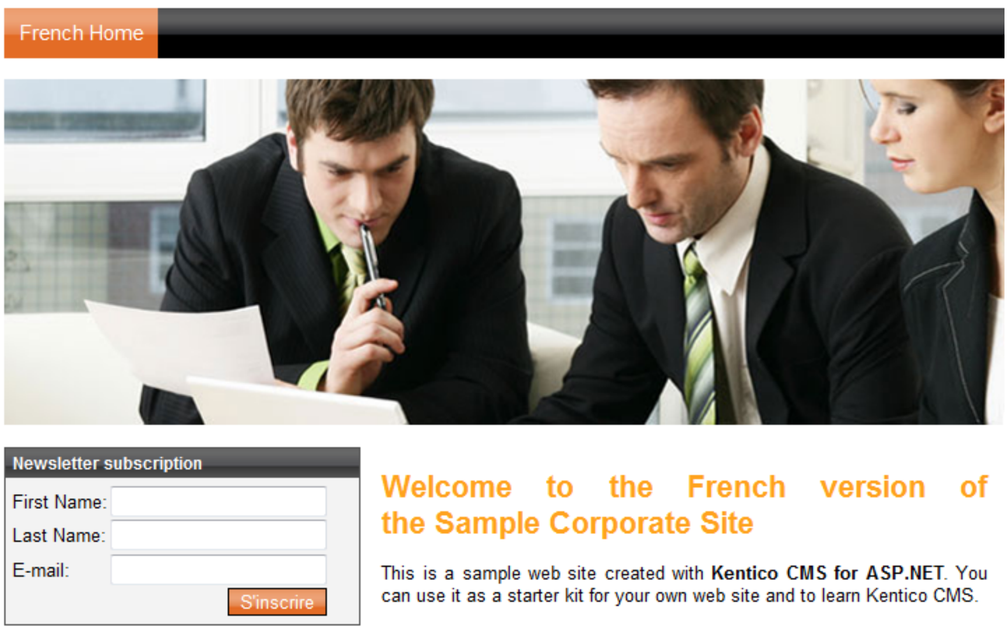
Click **Save**.

6. Change the heading text in the **MainContentText** web part to "Welcome to the French version of the Sample Corporate Site".



Click **Save**.

7. Sign out. Now, you can see the live site in French version:



8. As you can see, you can switch between languages using the language selection links at the top. They are displayed automatically using the **Language selection** web part.

9. Now sign in as administrator to **Site Manager and go to Settings -> website**. Choose the Corporate Site in the drop-down list and set the value **Combine with default culture** to **true**. Click **Save**. Sign out and see the French version of the live website now:

As you can see, the documents (menu items) that are not translated to French, are now displayed in the default culture (English).

Creating content for another language version

Every time you create content for another language version, please be sure you "translate" at least the **root** document and the **home page**. Otherwise, the website will not be displayed correctly.

The documents that are not translated yet are displayed with a little cross icon on the right so that you know what needs to be translated.

7.12.3 Configuring multilingual and RTL UI

Kentico CMS allows you to manage content in any language, including double-byte (eastern) languages, such as Chinese, and right-to-left languages (such as Hebrew or Arabic). All content is stored and published in UNICODE.

Translating the administration interface

If you want the administration interface to be displayed in a different language or at least with different culture settings (e.g. calendar and numeric format), go to **Site Manager -> Development -> UI cultures** and create a new UI culture:

- **UI culture name**: Hebrew (example - you can use any other culture)
- **UI culture code**: he-IL (example - you can use any other culture code)

Click **OK**.

Now create a copy of file **<web project>\CMSResources\cms.resx** under name **cms.he-IL.resx** (generally **cms.<culture code>.resx**) in the same folder and translate the strings.

Please note: when you make changes to the .resx file, you need to restart the web application using the **Site Manager -> Administration -> System -> Restart application** button so that the changes are updated in the user interface.

Modifying standard strings

If you want to modify some text in the user interface (including web parts), you can create a **custom.resx** file and put your strings to this file. The name of the string must be the same as in the **cms.resx** file. This procedure allows you to modify the strings without worrying that your changes will be overwritten during an upgrade to a newer version.

If you need to customize strings in non-english resource file, your custom file must use a name like **custom.fr-fr.resx** for French.

How to add your own strings

If you need to translate your own website strings, such as form labels or other static text into several languages, you can create custom string in **Site Manager -> Development -> UI cultures** either in English or only in some specific language. Please be sure to check the **Custom string** box in this case so that the string is automatically exported with your website.

Then, you can retrieve the string using the **CMS.GlobalHelper.ResHelper.GetString** method in your custom code.

Please see chapter Localization Expressions to see how you can insert localized strings into text values throughout the CMS.

Priority of the resource strings

When looking for a localized strings, the system uses the following priority:

1. database (Site Manager -> Development -> UI Cultures)
2. custom.resx
3. cms.resx

It means if there are duplicate strings with same name in all three files, the system will use the one stored in the database.

To change the priorities, you can add the following key to your web.config:

```
<add key="CMSUseSQLResourceManagerAsPrimary" value="false" />
```

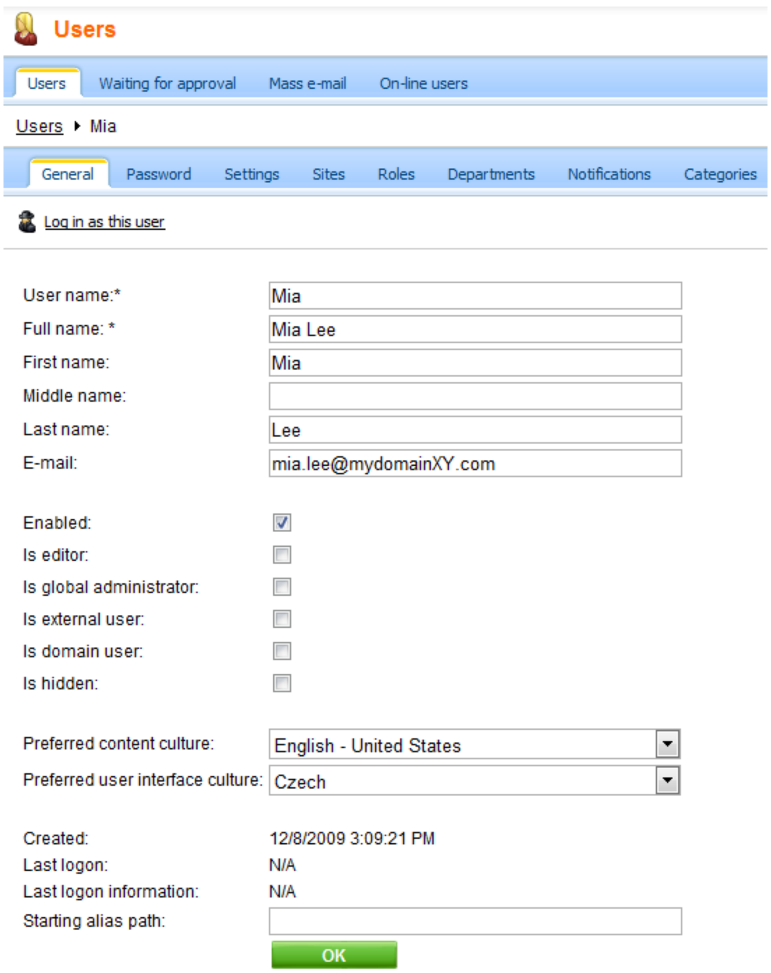
When the key is added, the priorities are the following:

1. custom.resx
2. cms.resx

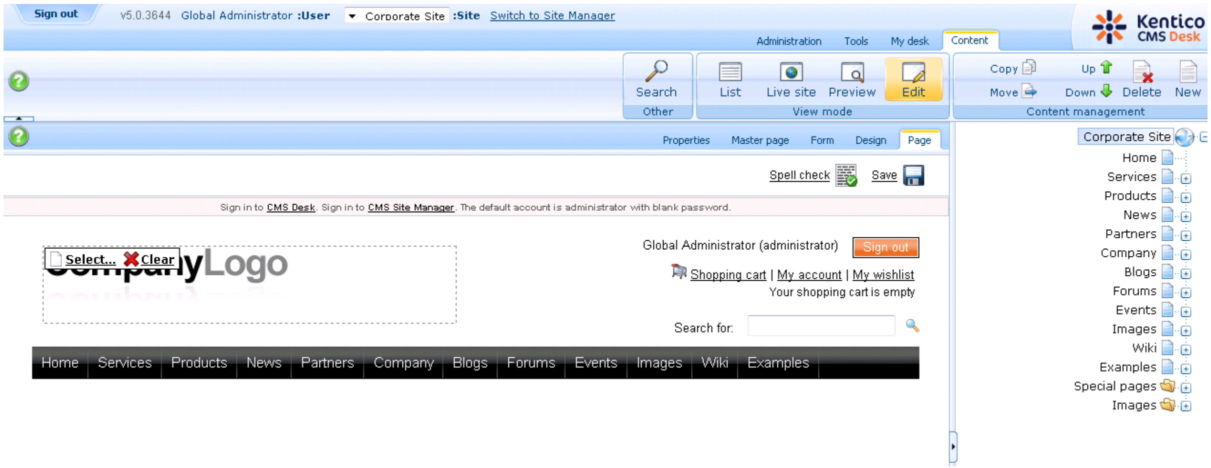
3. database (Site Manager -> Development -> UI Cultures)

Applying selected culture to the user interface

Now if you want to view the user interface in a different culture, you need to go to **Site Manager -> Administration -> Users ->** edit your user profile and set the **Preferred user interface culture** to the required value:



When you sign out and sign in again, the user interface is displayed with the new culture settings and translated strings (after you translate them):



7.12.4 RTL languages

If the website is displayed in language that uses right-to-left direction of written text (typically arabic languages), the CSS style of the BODY element is set to RTL (<body class="RTL">). You may need to add right-to-left specific CSS styles modifications.

RTL in UI culture vs. content culture

The **user interface direction** (RTL or LTR) is driven by the **preferred UI culture** of the current user. The **content direction** is driven by the **preferred (content) culture** of the current user.

Example:

Original LTR style:

```
.xxx
{
text-align: left;
float: left;
border-right: solid 1px #cccccc;
}
```

RTL style:

```
.RTL .xxx
{
text-align: right;
float: right;
border-right: none;
border-left: solid 1px #cccccc;
```

```
}
```

Adding culture-specific fonts

If your culture uses specific fonts that are not available in the WYSIWYG editor, you need to configure it:

1. Open file <web project>\CMSAdminControls\FCKeditor\fckconfig.js in notepad.
2. Add your font names on the following line:

```
FCKConfig.FontNames = 'Arial;Comic Sans MS;Courier New;Tahoma;Times New Roman;
Verdana';
```

3. Save the file.
4. Clear the cache of your web browser (Internet Explorer: Tools -> Internet Options -> General -> Delete files..., check the "delete all off-line content" box and click OK).
5. Close the browser and sign in to Kentico CMS Desk again. Now you should see the new font(s) in the WYSIWYG editor's font list.

7.12.5 Languages and URLs

When you're using multilingual support, all language versions of the given document use the same URL (based on the alias path) by default. E.g. the home page always has this URL: `/Home.aspx`

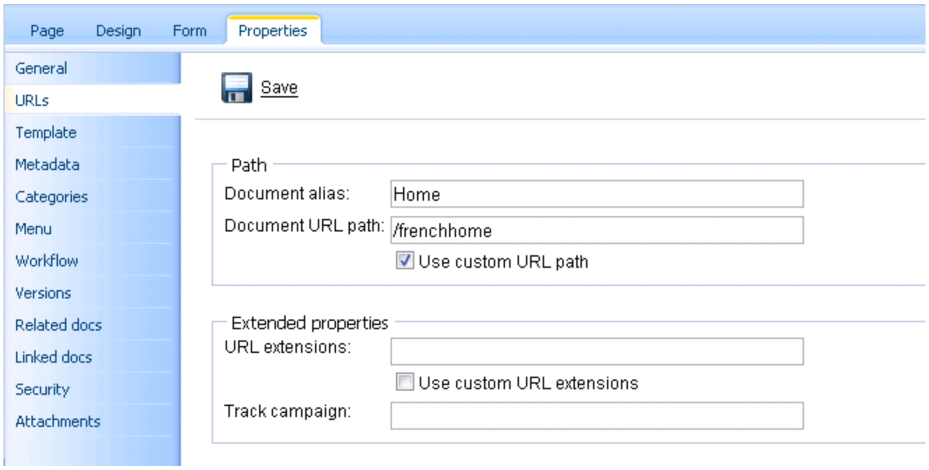
If you want to see the same page in French, you need to go to: `/Home.aspx?lang=fr-fr`

Once the language is changed, the selected language is stored in the visitor browser's cookie and when the user comes back to the */Home.aspx* URL (without any parameters), the French version is displayed to them.

Using custom URL path for different culture versions

If you want the French home page to have different URL than the English version, you need to go to **CMS Desk -> Content** -> choose the French language, select /Home in the content tree and click **Properties -> General**.

Check the **Use custom URL path** box and set the **Document URL Path** value to: **/frenchhome**



Now sign out, switch to the English version and go to URL: <web project>/frenchhome.aspx page
The website culture is automatically switched to French and you're displayed with French version.

You can also find some information on using Wildcard URLs on multi-lingual sites in this topic.

7.12.6 Default language selection

When a user first comes to the website, their content language is selected and stored in a cookie. You can specify for every domain and domain alias which culture should be used by default.

Go to **Site Manager -> Sites** and edit site Corporate Site. On the **General** tab, you can see the **Default visitor culture** field. You can choose either some particular language that will be used by default or you can choose the option **(Automatic)** that displays the content based on the preferred language of visitor's web browser (in Internet Explorer, you can set the default language in Tools -> Internet Options -> Languages). It automatically selects the nearest content culture that is available for the given website.

Language selection based on the domain

You can configure the system to use different default languages based on the current domain. If you use the following domains:

- example.com (main domain)
- example.de (domain alias)
- example.fr (domain alias)

you need to set the default cultures English, German and French to the domains/domain aliases (respectively). Now, when a visitor comes first to e.g. *example.fr* domain, the content is automatically displayed in French.

7.12.7 Localization Expressions

If you need to supply a localized value into web part property or to some other setting where localization expressions are supported, you can use expressions in following formats:

Format	Description	Sample Value

Basic format: `{$key$}`	Displays value of the given string specified in the Site Manager -> Development -> UI Cultures section or in the resx files in the CMSResources folder.	{$myform.firstname$}
`In-place localization:` `{$=default_string\|culture_code=translation\|culture_code=translation\|etc.$}`	Displays the strings defined in the expression. On the left, you can see an example that displays Hallo for German culture, Ciao for Italian and Hello for all other cultures (default value).	{$=Hello\|de-de=Hallo\|it-it=Ciao$}

See also: Appendix A - Macro expressions

7.12.8 Translation management

7.12.8.1 Overview

The translation management features brings various improvements aimed to facilitate website localization. There is a large number of improvements planned to be implemented. Due to the complexity, only the first implementation phase can be found in this version of Kentico CMS. The full functionality is supposed to be implemented in one of the following versions.

What is already available?

- Culture-dependent workflow scopes - each language version can have its own workflow defined
- Translation status overview on the List tab - you can see which language versions of documents under the selected node are translated, missing or outdated
- New Properties -> Languages tab - for each document, this tab displays information about all language versions of the current document
- Language-bound editors - you can specify which users can edit particular language versions of documents

7.12.8.2 Culture-dependent workflow scopes

Workflows can now be defined separately for each language version of your documents by using culture-dependent workflow scopes. A workflow scope defines which documents should a particular workflow be applied to. If you are not familiar with the workflow functionality of Kentico CMS, please refer to the **Workflow and versioning** chapter of this guide and especially this topic, where the process of defining a workflow is described step-by-step.

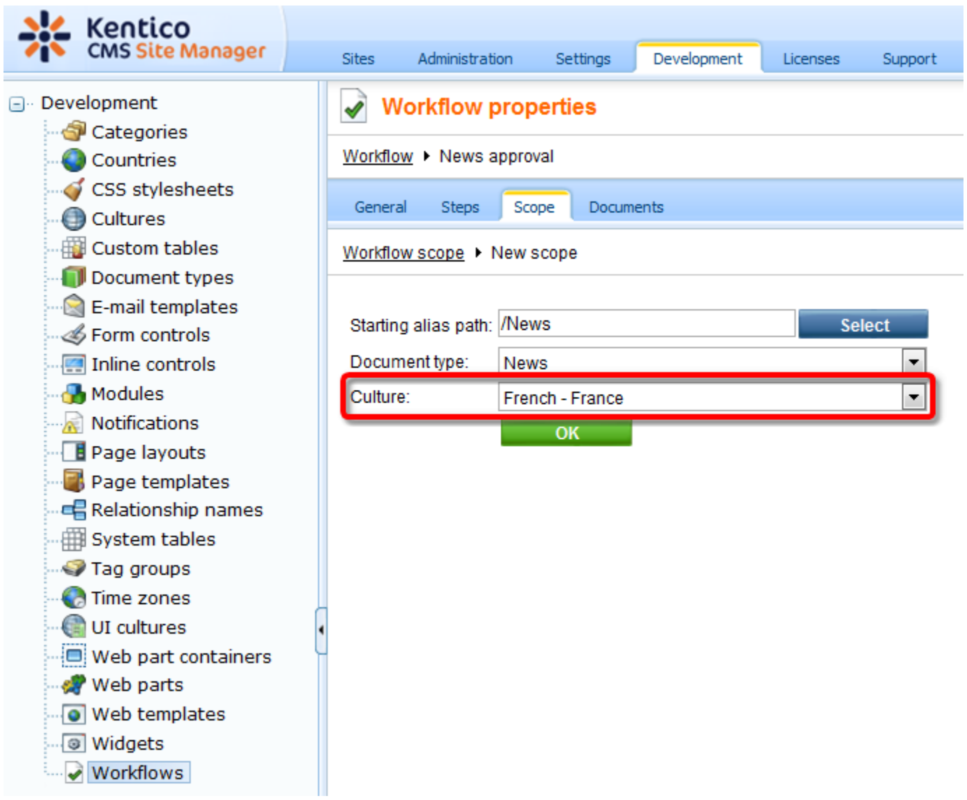
When defining a workflow scope, you can now specify the **Culture** property. Using this property, you can specify that the workflow will be applied only to the specific language versions of the documents.

The three workflow scope properties work as the following:

- **Starting alias path** - the workflow will be applied only to documents stored under the entered alias path
- **Document type** - the workflow will be applied only to documents of the selected document type
- **Culture** - the workflow will be applied only to the selected language versions of documents specified by the two properties above; choose (all) for the workflow to be applied to all cultural versions

Workflow scopes are applied with the following priorities (from highest to lowest):

1. Scope with specified document type and culture
2. Scope with specified document type
3. Scope with specified culture
4. Scope without specified document type and culture



7.12.8.3 Translation status overview

CMS Desk's **List** tab provides a useful overview of language versions of the documents. By selecting a node from the content tree and switching to this tab, you can see a list of documents placed under the selected node.

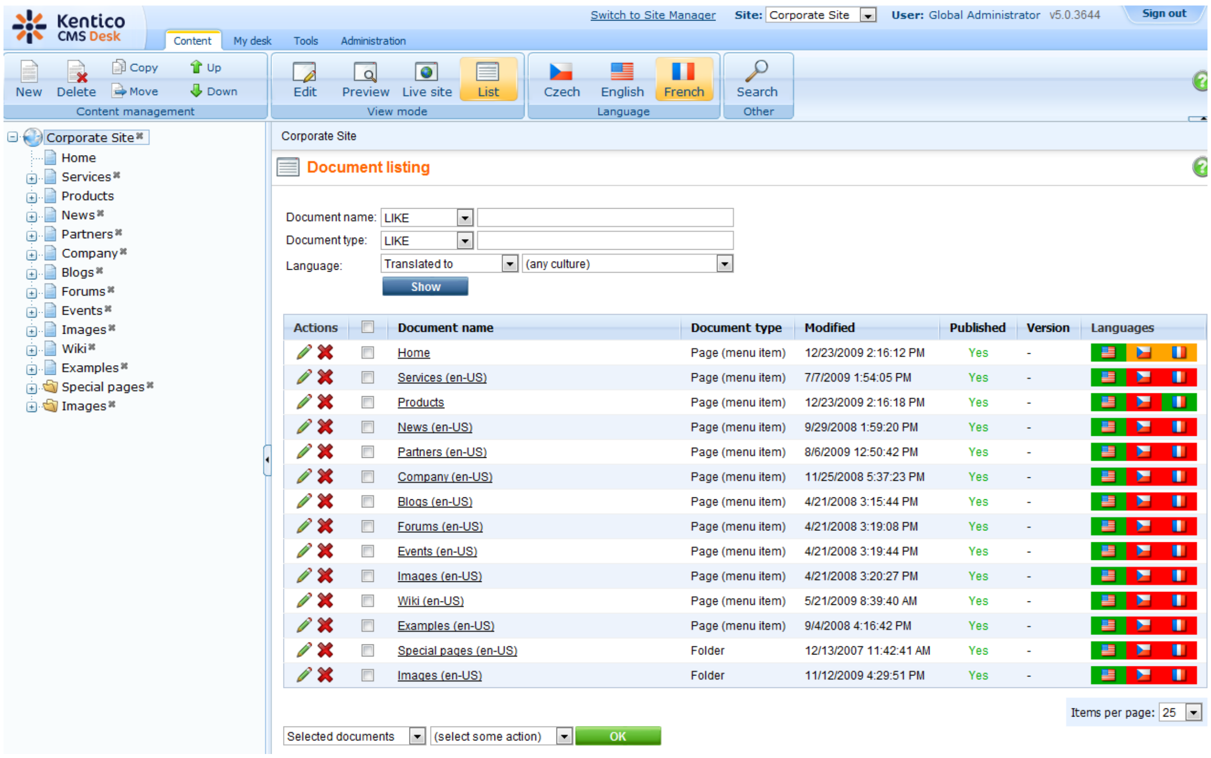
As you can see in the screenshot below, there is the **Languages** column with flags on different colored backgrounds. This column is displayed only on multi-language sites. The flags represent the particular language versions of each document.

The colors indicate the following translation statuses with respect to the default site culture:

- **Green - Translated** - the document is translated and up-to-date
- **Orange - Outdated** - the document is translated but outdated, which means that the default language version has been modified (or published when using workflow) more recently than the translated version
- **Red - Not available** - the document's version in the language does not exist

If you click a **Translated** or **Outdated** flag, you are redirected to the **Edit -> Page** tab of the appropriate language version of the document. If you click a **Not available** flag, you are redirected to the language version's creation dialog.

Another thing you may notice in the screenshot below is the **(en-US)** string attached to some document's names. The **Name** column displays names of the documents in the currently edited culture (which is French in the screenshot). If the document in this language version **does not exist**, the column displays the name from the **default culture** with the default culture code appended in brackets.

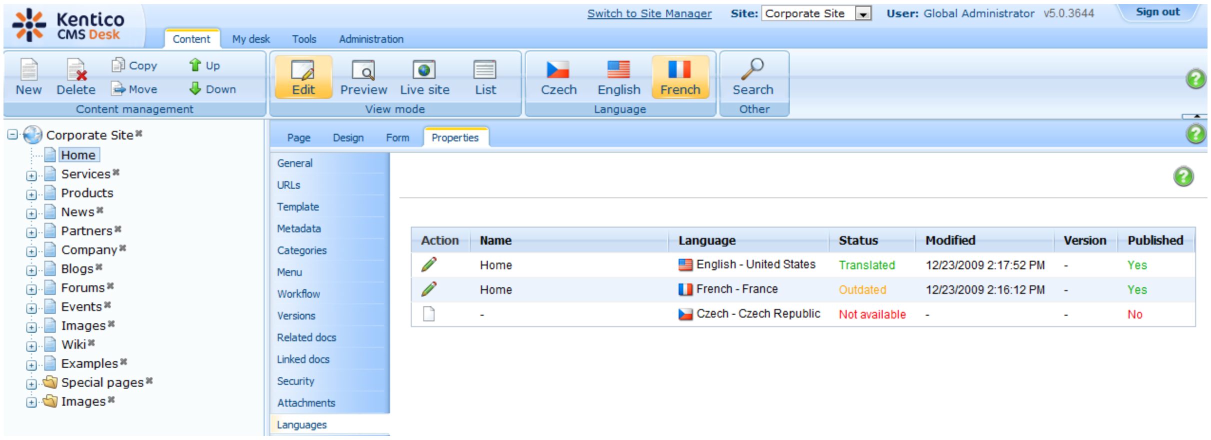


If you hover a flag with the mouse cursor, an info box will be displayed, giving additional information about the language version.

The Languages tab

The same statuses can be shown separately for each document. You can view them by selecting the document from the content tree and switching to the **Properties -> Languages** tab.

By clicking the **Edit culture version** ( ) icon, you can be redirected to the **Edit -> Page** tab of the selected culture version. By clicking the **Add new culture version** ( ) icon of a not-available version, you can proceed to creating the new culture version of the document.



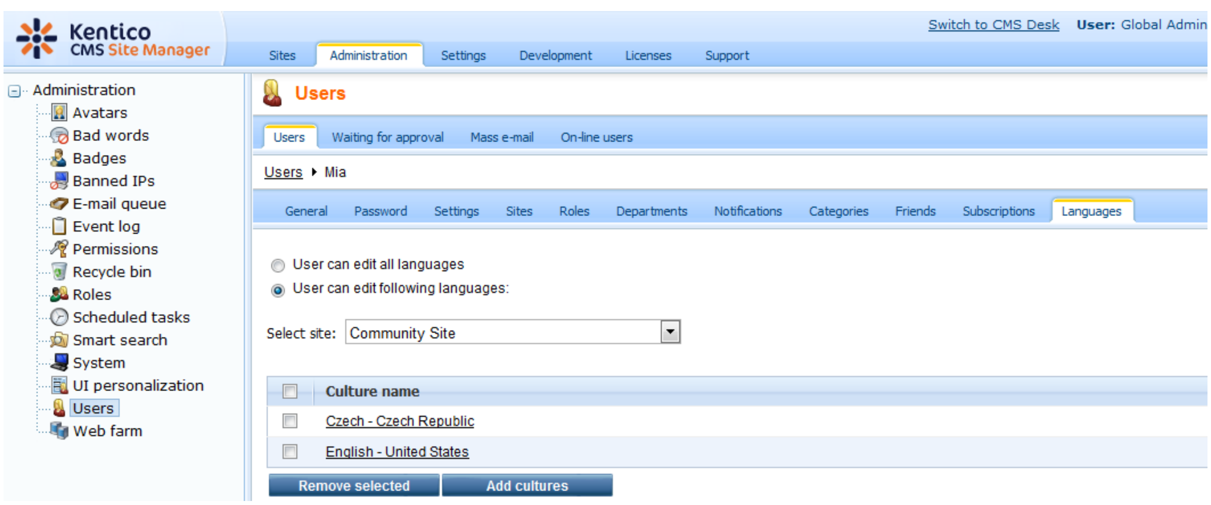
Hiding the Languages tab

The **Languages** tab can be displayed or hidden to members of particular roles. This can be set up using UI personalization as described here.

7.12.8.4 Language-bound editors

You can also specify who will be able to edit which language versions of documents. This setting can be done separately for each user. If you go to **Site Manager -> Administration -> Users**, choose to **Edit** ( ) a particular user and switch to the **Languages** tab, you have the following two options:

- **User can edit all languages** - if selected, the currently edited user can edit documents in all language versions of all sites in the system
- **User can edit following languages** - if selected, you can specify which language versions can be edited by the user by checking the check-boxes in the list of language versions; this can be set separately for each site in the system using the **Select site** drop-down list

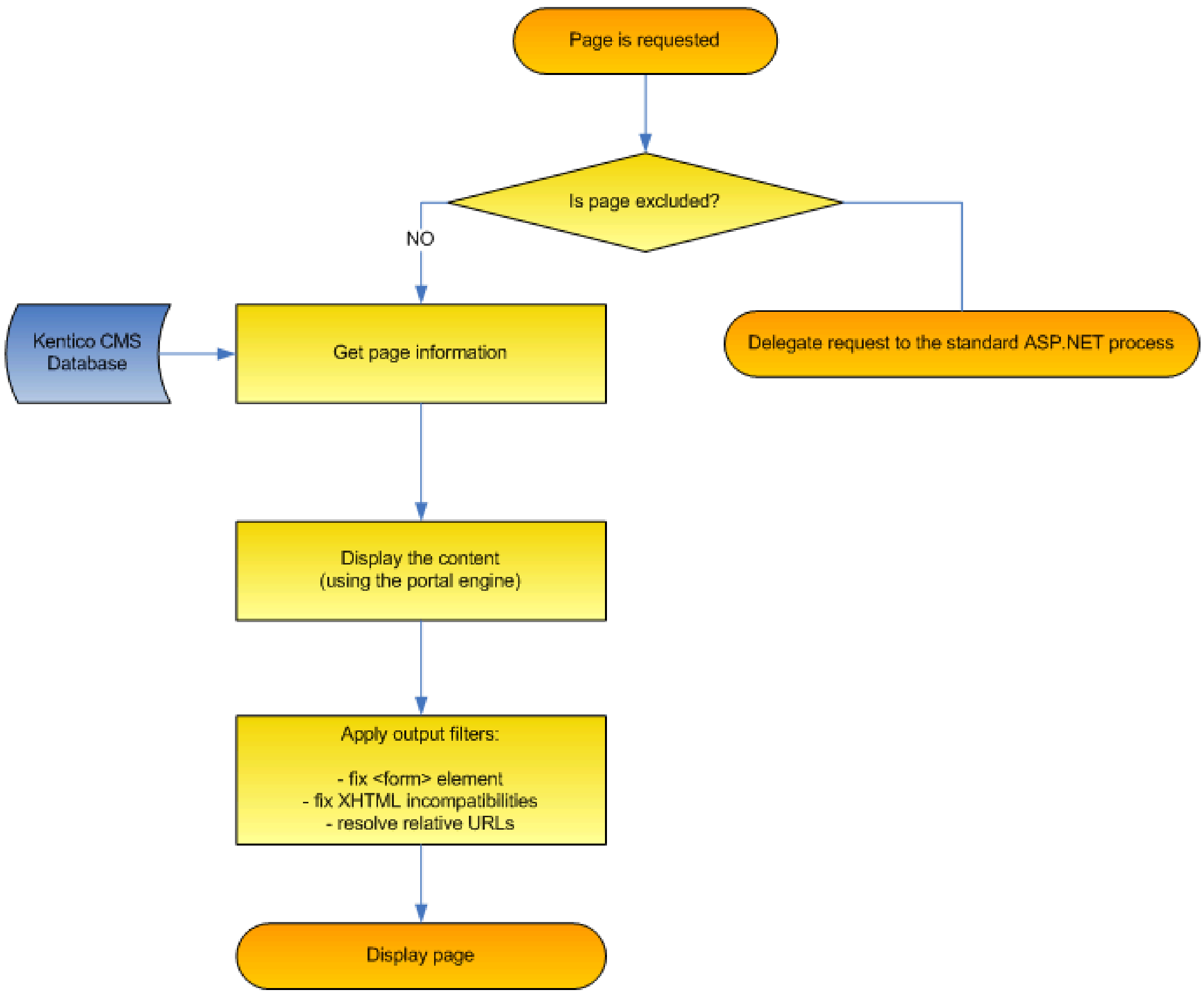


7.13 Page processing and URLs

7.13.1 Overview

Kentico CMS processes URLs using a **URL rewriting engine**. This engine ensures the displaying of the correct page based on the required friendly (smart) URL. After the page is processed by the URL rewriting engine, it is run through the output filters that ensure additional changes in the rendered HTML code.

The following figure shows the page processing step-by-step:



7.13.2 URL rewriting

Kentico CMS uses a system of friendly URL addresses. It allows you to use URLs like:

`http://www.example.com/products/kentico-cms.aspx`

instead of

`http://www.example.com/products.aspx?id=527`

Every document has its own URL. When multiple languages are used, the document is recognized either using its **URL path** (if specified) or using a combination of **alias path and preferred culture** that is stored in a cookie.

URL processing

1. The system gets the incoming request **`http://www.example.com/products/kentico-cms.aspx`** and looks up the website based on the domain name (either on the main domain name or domain aliases). If it doesn't find any running website, it displays the page **`/cmsmessages/invalidwebsite.aspx`**. If the domain name with the required port number is not found, it also tries to find a site with the same domain name without any port number.

2. Then it looks up a document with alias path equal to **`/products/kentico-cms`** and in the current culture. It can optionally also search for the default culture version of the same document in case the required culture version is not available - this can be set in **Site Manager -> Settings -> Web site -> Combine with default culture**.

3. If such a URL path is not found, the system tries to find a document with URL path **`/products/kentico-cms`**.

4. If no document is found for the requested URL, the system doesn't process the request and the web server displays the standard 404 – Page not found error.

5. If the requested document is found, the URL rewriting engine looks up which page template should be used to display it and calls the appropriate page template like this: **`/products.aspx?aliaspath=/prodcuts/kentico-cms`**. The page template is responsible for the displaying of the document. If the document doesn't have any page template specified, the URL rewriting engine tries to find and use the page template of the parent document. If the page is managed by the portal engine, the page called is **`/cmspages/portaltemplate.aspx`**, which is a page that renders the page from web parts.

The system of "friendly" or "smart" URLs provides several benefits:

- They are easy to remember and easy to write into the browser address bar.
- They are friendly towards search engines (SEO friendly).
- They show users where they are located on the website.
- You can easily send the URL of the document to your friends and they will see the same page with the particular document.

7.13.3 Multiple document aliases

In versions prior to Kentico CMS 4.0, you could have two URLs leading to one document. Both of these could be set in CMS Desk, on the **Properties -> URLs** tab of each document:

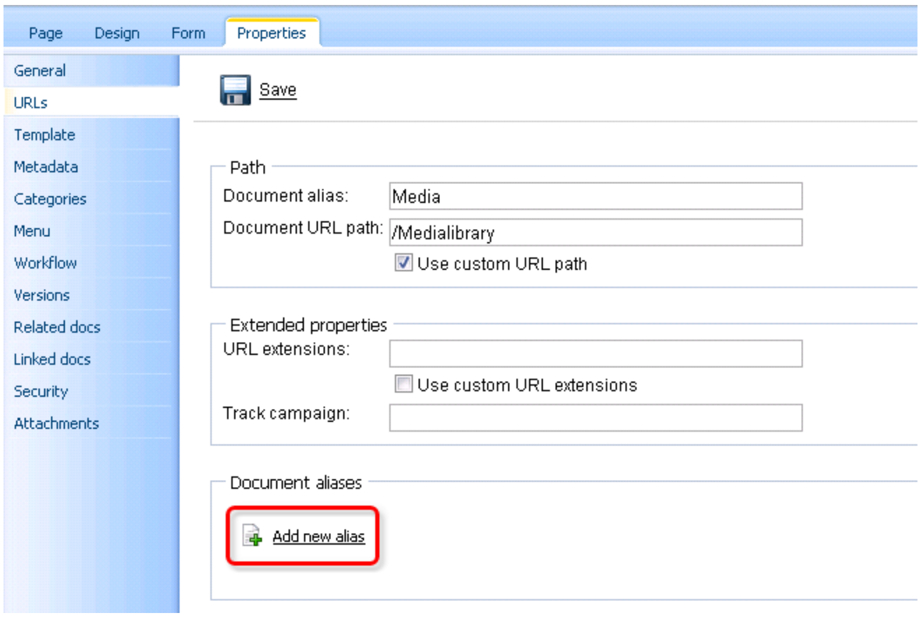
- **Document alias** - this is the unique name of the document in the given section of the website; if *Media* is the value, then *<domain>/Media.aspx* is the URL under which the document can be accessed by default

- **Document URL path** - this is the URL alias that could be used in the versions preceding 4.0; if you enter */Medialibrary*, the page will also become accessible through *<domain>/Medialibrary.aspx*

Kentico CMS 4.0 brings the possibility of having an unlimited number of aliases for each document. Adding new document aliases can be done the following way:

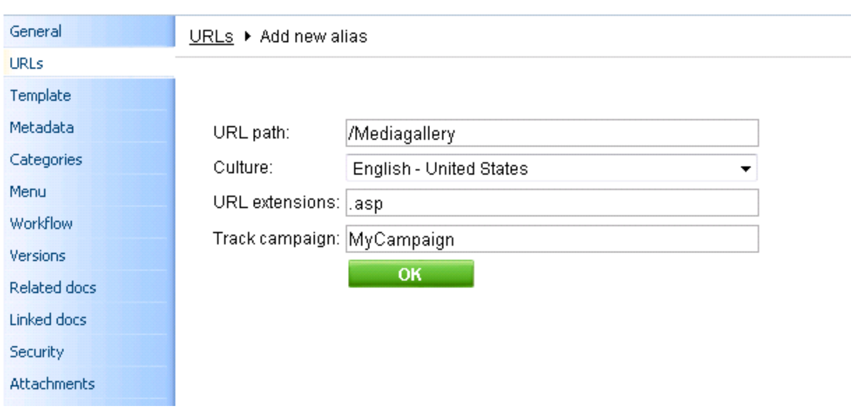
1. Sign in to CMS Desk, select some document from the content tree and switch to its **Properties -> URLs** tab.

2. Click the **Add new alias** link in the **Document aliases** section.



3. Enter the following details:

- **URL path** - URL of the alias
- **Culture** - cultural version of the document that will be displayed when accessing the page through this URL
- **URL extensions** - additional supported extensions of the URL; for these to work, you will have to configure the system as described in Configuration of custom URL extensions; optional field
- **Track campaign** - name of a web analytics campaign that will be created on the first access through this URL alias and that will be used for tracking access to the document through the alias; optional field



Click **OK** to create the alias.

4. Now if you switch back to the **Properties -> URLs** tab of the document, you should see the newly

created alias present in the list in the **Document aliases** section, as depicted in the screenshot below. Like this, you can add an unlimited number of aliases.

Wildcard URLs

Wildcard URLs can also be used in these additional document aliases. For more information on wildcard URLs, please refer to the Wildcard URLs chapter of this guide.

7.13.4 URL format and configuration

Defining the extension of URLs

The URLs can use various extensions. By default, all URLs end with **.aspx**, such as: `http://www.example.com/products/kentico-cms.aspx`

You can also use **custom extensions**, such as `.htm`, `.html` or any custom extension. Alternatively, you can even use **URLs without extensions**, such as `http://www.example.com/products/kentico-cms`. However, in this case, you need to configure the system as described in Configuration of custom URL extensions.

Excluding URLs from the CMS engine

If you need to add your own pages to the website, you may need to exclude them from CMS engine processing. You can do that by adding the page URL (without extension) to the **Site Manager -> Settings -> URLs and SEO -> Excluded URLs** value (you can enter several URLs separated by a semicolon (;)).

Forbidden URL characters

The URLs (alias paths and URL paths) cannot contain some special characters. By default, these are the forbidden characters:

\ / : * ? " " < > | & % . ' # [ ] and the space character.

You can add additional forbidden characters by entering them (without any separator) into the **Site Manager -> Settings -> URLs and SEO -> Forbidden URL Characters** setting field.

The forbidden characters are removed by default. You can specify the character that will replace forbidden characters in the **Site Manager -> Settings -> URLs and SEO -> Forbidden Characters Replacement** setting field.

Using URL Prefix

If you need to add some prefix to all URLs (e.g. for search engine optimization), you can specify it in the **Site Manager -> Settings -> URLs and SEO -> Default Url Path Prefix** field.
The URLs will then look like this: `http://www.example.com/myprefix/products/kentico-cms.aspx`

Automatic creation of new document aliases

If you check the **Site Manager -> Settings -> URLs and SEO -> Remember original URLs when moving documents** check-box, new document aliases will automatically be created when a new extension or URL path is set.

URL related settings

The above configuration tasks can be performed in **Site Manager -> Settings -> URLs and SEO**. The following table contains these settings and provides descriptions for them:

<table>
<tr><td>Forbidden URL characters</td><td>List of additional characters that can't be used in the document alias and in the file system. The following characters are forbidden by default: ' \ /:*? “ <>|& %.'#[]+ =</td></tr>
<tr><td>Forbidden characters replacement</td><td>Character that will be used as a replacement of forbidden characters in the document alias and in the file system.</td></tr>
<tr><td>Friendly URL extension</td><td>Specifies the extension of friendly URLs. The extension should be preceded by a dot when entered, such as: .aspx or .html. When you omit this value, the friendly URLs will be the same as the alias path (e.g. /products/nokia).
Please note: The system of “friendly” or “smart” URLs provides several benefits. They are easy to remember and easy to write into the browser address bar. They are search engine friendly (SEO friendly). They show users where they are located on the website. You can easily send the URL of the document to your friend and she will see the same page with the particular document.</td></tr>
<tr><td>Files friendly URL extension</td><td>Specifies the extension for files that will be used in friendly URLs, such as ".aspx" or ".html".

Example: getfile/<node alias>/myimage.aspx

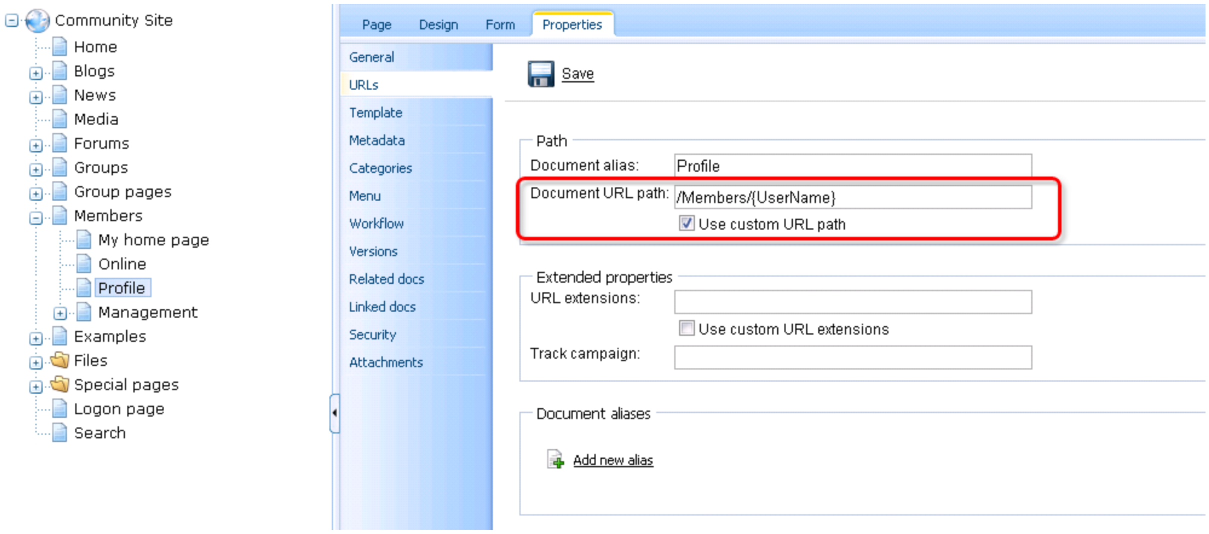
When you omit this value, the file friendly URLs will end with no extension: getfile/<node alias>/myimage</td></tr>
<tr><td>Excluded URLs</td><td>List of URLs (without domain) that should be excluded from the CMS engine. You can enter several paths separated by a semicolon (;).</td></tr>
<tr><td>Default URL path prefix</td><td>Defines a default URL path prefix that will be used for all URLs of the content pages. This prefix is rewritten to ‘urlpathprefix’ query string parameter.</td></tr>
<tr><td>Use name path for URL path</td><td>If checked, this key indicates that a document's name path will automatically be copied to its URL path.</td></tr>
</table>

Use permanent URLs	If enabled, URLs of documents and document attachments will be generated in permanent format; if disabled, friendly URLs will be used. Learn more at Linking pages and files.
Remember original URLs when moving documents	Determines if new document aliases should be created when a new document URL path or extension is set.
Google sitemap URL	URL that will be automatically rewritten to the physical location of the Google Sitemap. The physical location is ~/ *CMSPages/GoogleSiteMap.aspx*. More information can be found at Google Sitemap.
Allow permanent (301) redirection	If enabled, the system uses permanent (301) redirection instead of standard (302) redirection for SEO purposes.
Use URLs with trailing slash	Specifies how URLs with trailing slash should be handled. Possible options: Leave the URL as is, Always use URLs with trailing slash, Always use URLs without trailing slash.
Redirect document aliases to main URL	If enabled, every document always has only one valid URL and other aliases are redirected to this main URL, for SEO purposes.
Redirect invalid case URLs to their correct versions	Checks the URL letter case, and if configured, redirects to their correct version. Options: Do not check the URLs case, Use the exact URL of the document, Redirect all requests to lower case URLs, Redirect all requests to upper case URLs.

7.13.5 Wildcard URLs

Wildcard URLs provide a way of loading content dynamically depending on the page URL. You can find an example of how this works on the sample Community starter site. The **Members -> Profile** page uses wildcard URLs to display user profiles. As you can see, it is only one single page that can display profiles of various site users.

How is it achieved? If you go to **CMS Desk**, select the **Members -> Profile** page from the content tree and switch to its **Properties** tab, you should see **/Members/{UserName}** in the **Document URL path** field. The **{UserName}** part of the URL is the actual wildcard.



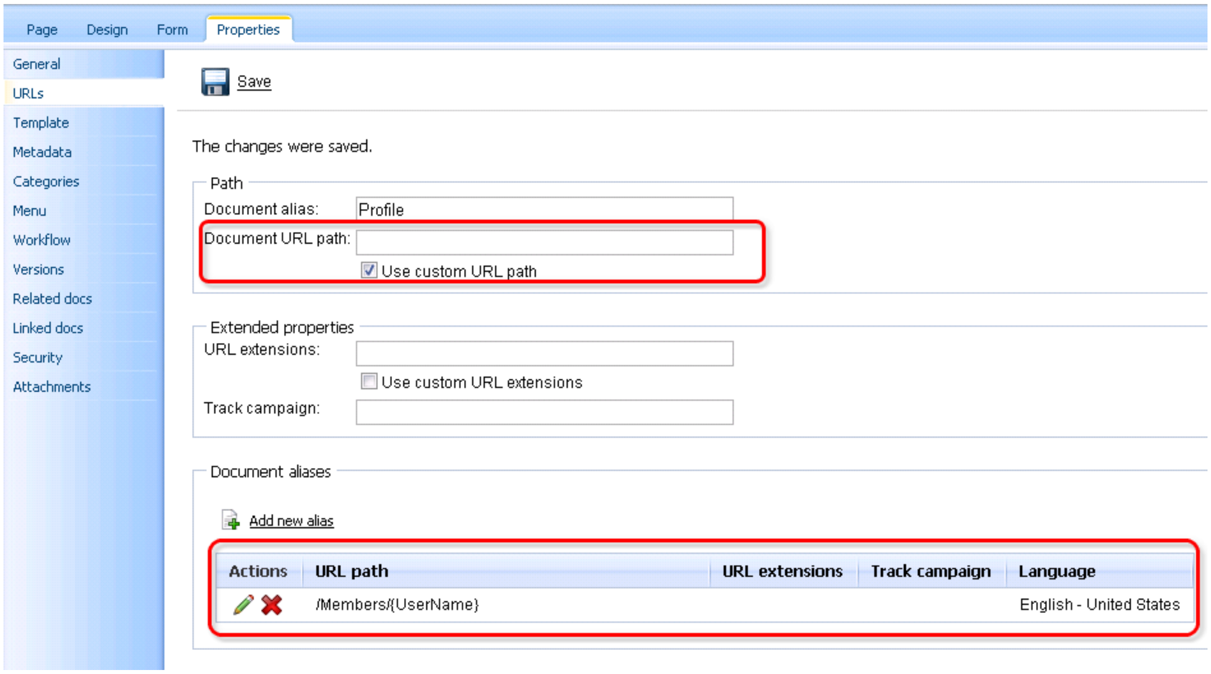
If you type **<domain>/Members/David.aspx** into your browser, the **Members -> Profile** page will be displayed and the wildcard part of the URL (David) gets translated into a query string parameter, so that the translated URL looks like this: **<domain>/Members/Profile.aspx?username=David**. As you can see, the name of the parameter is taken from the name of the wildcard, while the value is the matching part of the entered URL. The **User public profile** web part that is used on the page recognizes the username parameter in the rewritten URL and displays David's profile.

Using wildcard URLs on multi-language sites

The **Document URL path** is unique for each language version of a document. Because of this fact, you may encounter problems when referring to a page using a wildcard URL on multi-lingual sites. Let's explain the situation using the following example:

On the sample **Community Starter site**, the **Members/Profile** page has its Document URL path set to **/Members/{UserName}**. If you created a version of this page in another language, its Document URL path would get changed to **/Members/{UserName}-1** automatically. This happens because the Document URL paths need to be unique.
Now let's presume that you have the following link leading to the page: **<domain>/Members/David.aspx**. In the original version, it works fine. But if you tried to click the link in the second language version, you would get an error because the URL in this language version would be **<domain>/Members/David-1.aspx**.

If you want to keep such links functional in all language versions, you will need to define the **/Members/{UserName}** path via the **Document aliases** section, as described in the Multiple document aliases topic. When creating the alias, select **(all)** in the **Culture** drop-down list. At the same time, you must erase the value from the original document's Document URL path. As it has higher priority, URL paths defined in the Document aliases section would have no effect. The result should look as in the following screenshot:



Dots in wildcard URLs

You may encounter problems when a string containing a dot "." gets into the wildcard part of a URL. A typical example of this can be found on the **Members -> Profile** page of the sample Community Starter Site.

The page's Document URL path is set to **/Members/{UserName}**. Let's presume that you have the following user name: **jack.smith**. Then the user's profile page would be located at **http://<domain>/Members/jack.smith**. As the last part of the URL after the last dot ('smith' in this case) is understood as file extension, this URL would produce the 404 error in your browser.

To prevent this, registration on the sample Community Starter Site doesn't allow user names with dots. This is ensured by a validation of the **UserName** field in the **Registration form** alternative form of the **User** system table.

If you needed to allow dots in user names and use wildcard URLs with user names at the same time, you can achieve this by removing the validation and setting the page's Document URL path to something like **/Members/{UserName}/Profile**. In this case, the dot would be located in the middle of the URL and the URL should work fine.

7.13.6 Linking pages and files

Linking documents (pages)

If you need to create a permanent link to a document, you need to use a URL in the following format:

```
http://www.example.com/getdoc/016fad52-0d69-46d5-80dc-daec9173c0c7/Products.aspx
```

It's an equivalent of:

```
http://www.example.com/company/products.aspx
```

However, in the first case, the link keeps working even if you move the document to some other place.

The URL consists of the following parts:

<domain>/getdoc/<document GUID>/<document name><extension>

The <*document GUID*> value is a unique identifier of the document. You can find this value at **CMS Desk -> Content -> Properties -> General**, in the **Node GUID** field.

The <*document name*> value may contain any value - it's not used by the system and it's only used for search engine optimization. By default, the system uses the document name for this value.

Linking a specific language version of the document

If you need to link to a specific language version of the document, you need to use a URL in the following format:

```
http://www.example.com/getdoc/8FG7-84E394-FABD-5678/our-services/fr-fr.aspx
```

It displays the given document in French (if the document is translated). It's an equivalent of

```
http://www.example.com/company/our-services.aspx?lang=fr-fr
```

The URL consists of the following parts:

<domain>/getdoc/<document GUID>/<document name>/<culture code><extension>

Linking attachments

If you need to create a permanent link to a file uploaded as a document attachment, you need to use a URL in the following format:

```
http://www.example.com/getattachment/763c8921-be94-4610-99b4-25e8d3be5b08/logo.aspx
```

The URL consists of the following parts:

<domain>/getattachment/<file GUID>/<filename><extension>

The <*file GUID*> value is not the same as the document GUID. It's a GUID of the file in the *CMS_Attachment* table. You can find this GUID if you display the attachment in **CMS Desk** and view its URL.

The <*file name*> value can contain any text.

You can find more details on available parameters in the following topic: GetFile.aspx parameters.

7.13.7 GetFile.aspx parameters

The GetFile.aspx script is used in many cases to retrieve an uploaded file from the database. It is called whenever you use /getdoc, /getattachment or a direct URL based on the alias path of a cms.file document.

The GetFile.aspx script accepts the following URL parameters:

Parameter Name	Description	Sample Value
guid	Attachment GUID value	
nodeguid	Node GUID value	
versionhistoryid	Version history ID of the attachment. It can only be used together with the **guid** parameter.	
width	Resizes the image to a specified width (in pixels).	100
height	Resizes the image to a specified height (in pixels).	400
maxsidesize	Resizes the image to the specified size of the longest side (in pixels).	500
disposition	Indicates the output disposition of the file. You can use either **inline** (opens the file in the browser window if possible) or **attachment** disposition (opens the "Save or Open" dialog).	inline or attachment

7.13.8 Output filters

The output filters are applied to rendered HTML code. They make various changes to the HTML code before it is sent to the browser:

Form filter

The form filter fixes the issue with non-working postbacks on pages that use URL rewriting. It ensures that forms, dialogs and buttons will work correctly on Kentico CMS-managed pages.

You can exclude pages from this filter by adding their alias path to the **Site Manager -> Settings -> Output filter -> Excluded Output Form Filter URLs** field, separated by a semicolon (;).

XHTML filter

The XHTML filter fixes some XHTML incompatibilities. It closes unclosed tags, invalid <script> tags, etc.

You can exclude pages from this filter by adding their alias path to the **Site Manager -> Settings -> Output filter -> Excluded XHTML Filter URLs** field, separated by a semicolon (;).

XHTML errors may also be fixed in the WYSIWYG editor when they are saved. This can also be configured globally through the CMSWYSIWYGFixXHTML web.config parameter (supported values are "true" and "false").

Resolve filter

The resolve filter changes relative URLs in format `~/mypage1/mypage2.aspx` to `/application/mypage1/mypage2.aspx` (application running in a sub-folder) or `/mypage1/mypage2.aspx` (application running in the root). It changes only URLs inside **src** and **href** attributes.

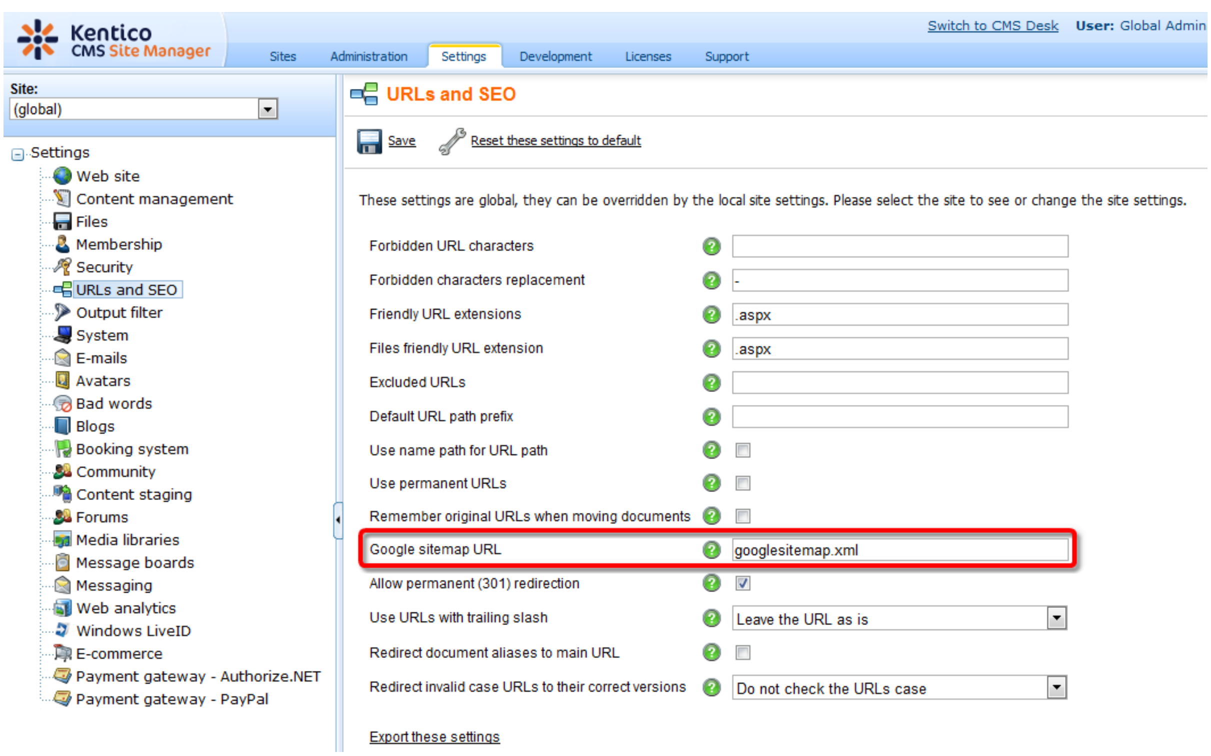
You can exclude pages from this filter by adding their alias path to the **Site Manager -> Settings -> Output filter -> Excluded Resolve Filter URLs** value, separated by a semicolon (;).

7.13.9 Google Sitemap

Kentico CMS comes with automatic support of the Google SiteMap Protocol. This is a protocol designed to help search engines with the indexing of your site and can improve your site's position in search engine results. Simply put, a Sitemap is an XML file where URLs of your site are stored. Detailed information about the SiteMap Protocol can be found at http://en.wikipedia.org/wiki/Site_map.

Kentico CMS creates your site's sitemap automatically. Physically, it is located at *<domain>/CMSPages/googlesitemap.aspx*. It can also be accessed through the URL specified in **Site Manager -> Settings -> URLs and SEO -> Google sitemap URL**. This value is set to *googlesitemap.xml* by default, which means that the site's sitemap can also be accessed through *<domain>/googlesitemap.xml*. This is the URL that you enter into google when registering a new sitemap.

Please note that if you want the sitemap to be accessible via a URL with the **.xml** extension, you need to configure your IIS to handle the **404 error** through Kentico's **handler404**, as described in the Installation and deployment -> Additional configuration tasks -> Custom URL extensions and extensionless URLs chapter. In this case, it is also necessary **not to set** the .xml extension to be processed by .NET.

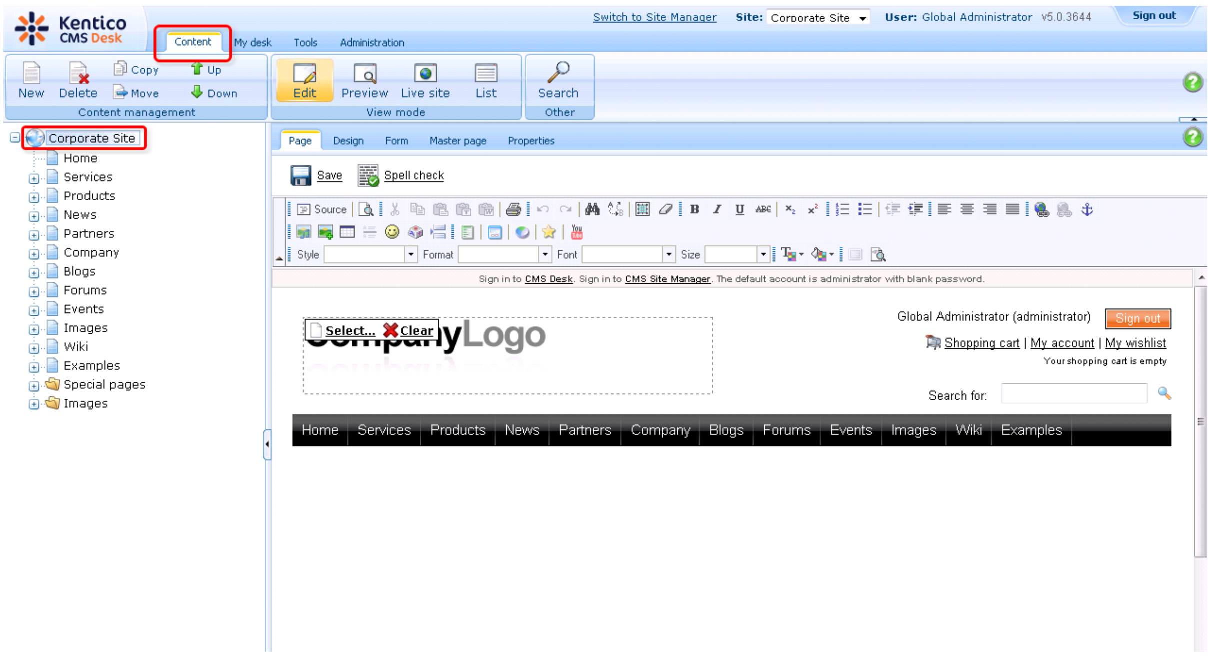


7.14 Rebranding

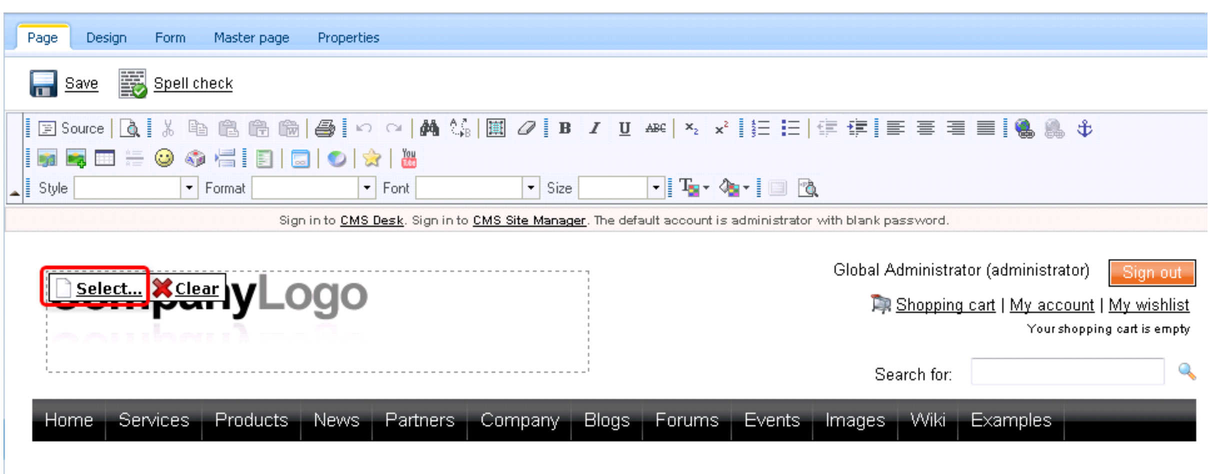
7.14.1 Changing a logo in the header

In this topic, you will learn how to change the header logo on the sample Corporate Site.

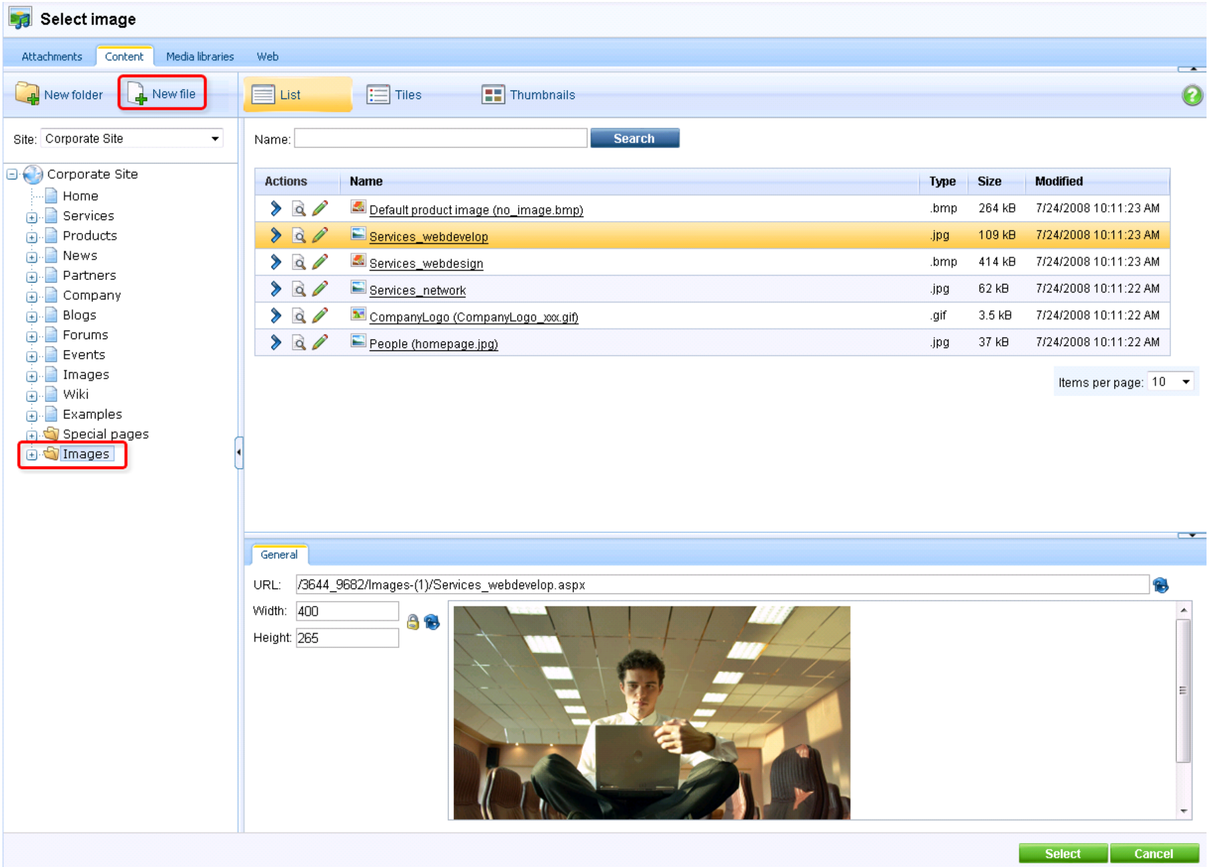
1. Go to **CMS Desk** -> **Content** and select the root of the content tree (**Corporate Site**).



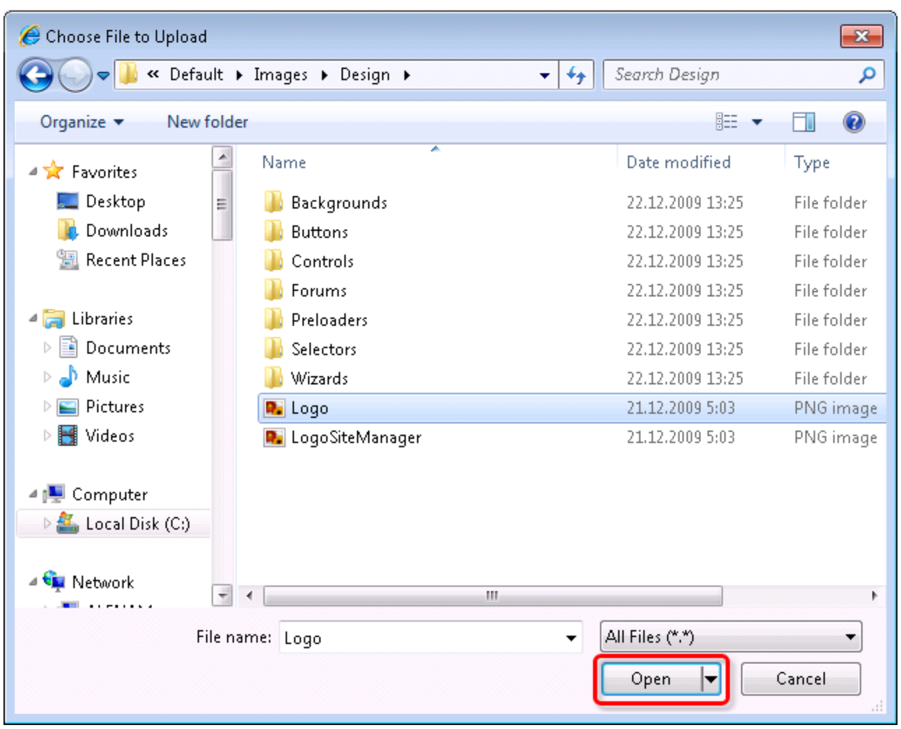
2. Click **Select ...** above the logo.



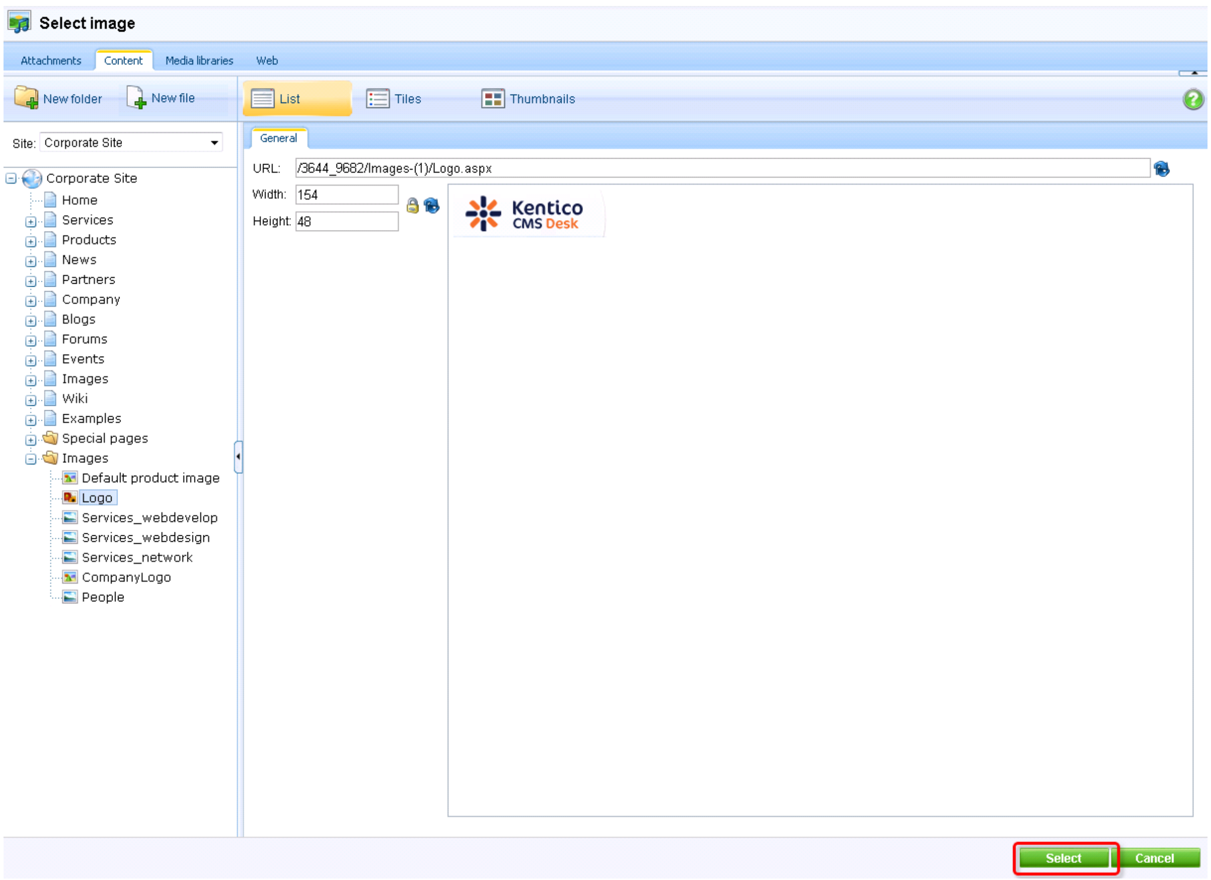
3. In the **Select image** dialog that pops up, select **Images** in the content tree and click the **New file** button above the content tree.



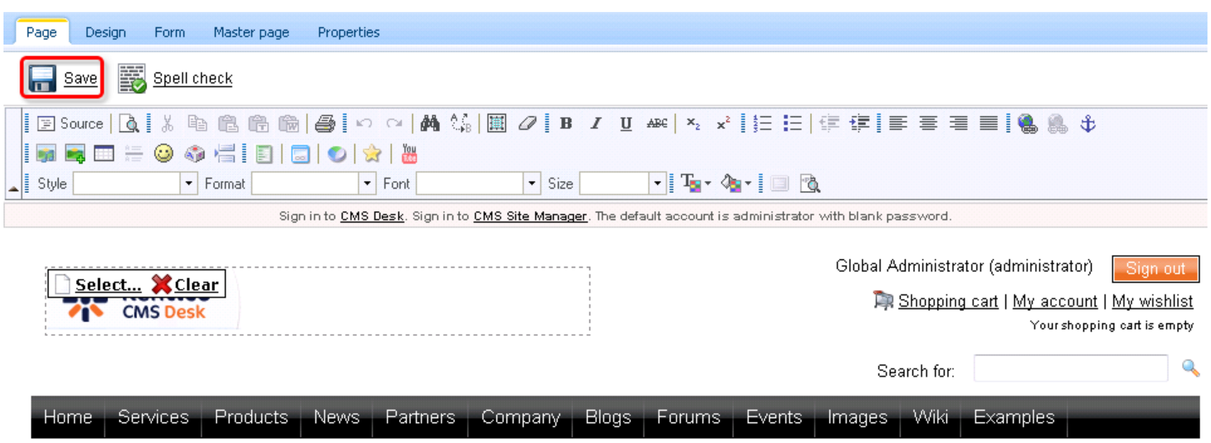
4. Find an image you want to upload and click **Open**.



7. The logo should be added to the Images folder. With the logo selected in the content tree, click **Select**.



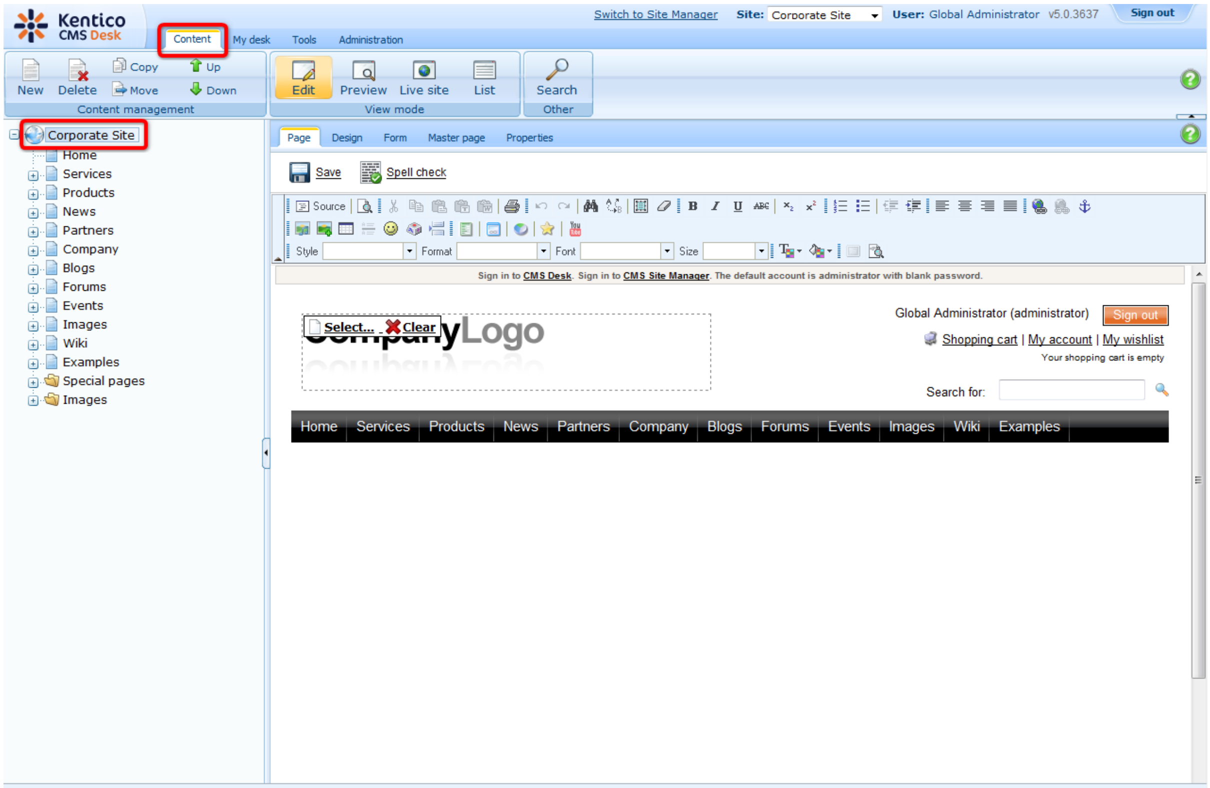
8. Now click **Save**.



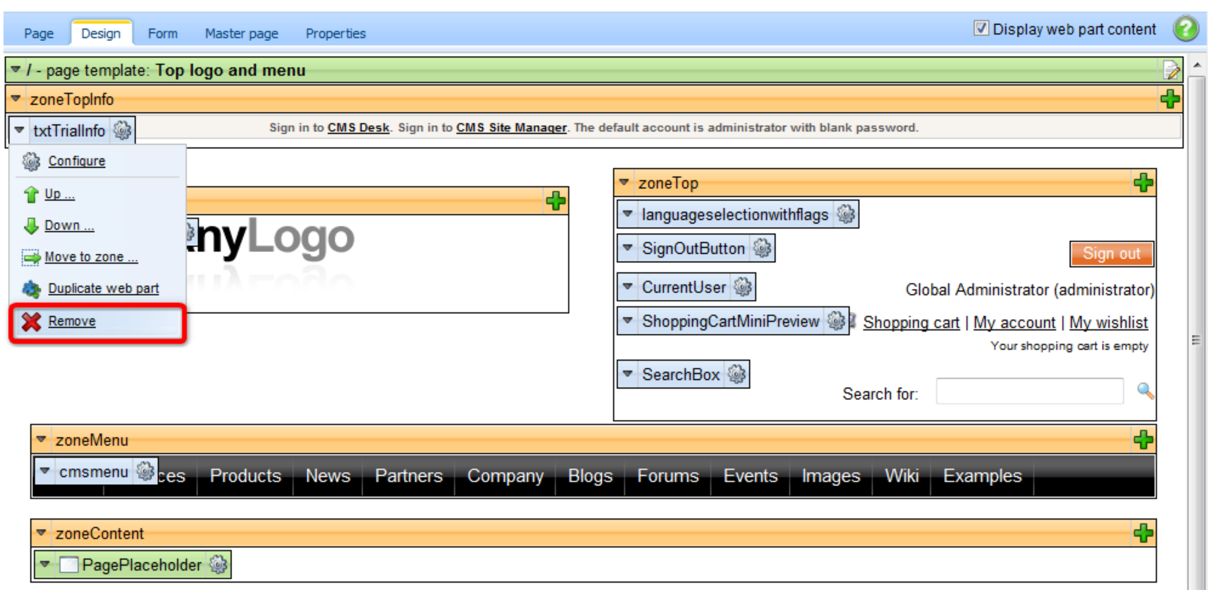
You've just publish a new logo on your website.

7.14.2 Removing the log-on bar

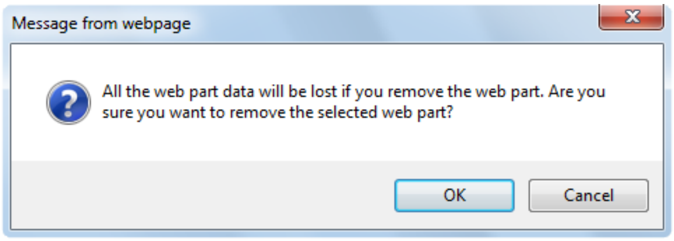
1. Go to **CMS Desk** -> **Content** -> **Corporate Site**.



2. Switch to the **Design** tab and right-clicked the **txtTrialInfo** webpart at the top-left and choose **Remove**.



3. Click **OK** to remove the bar from your website.



That's how you remove the signing-in bar from your website.

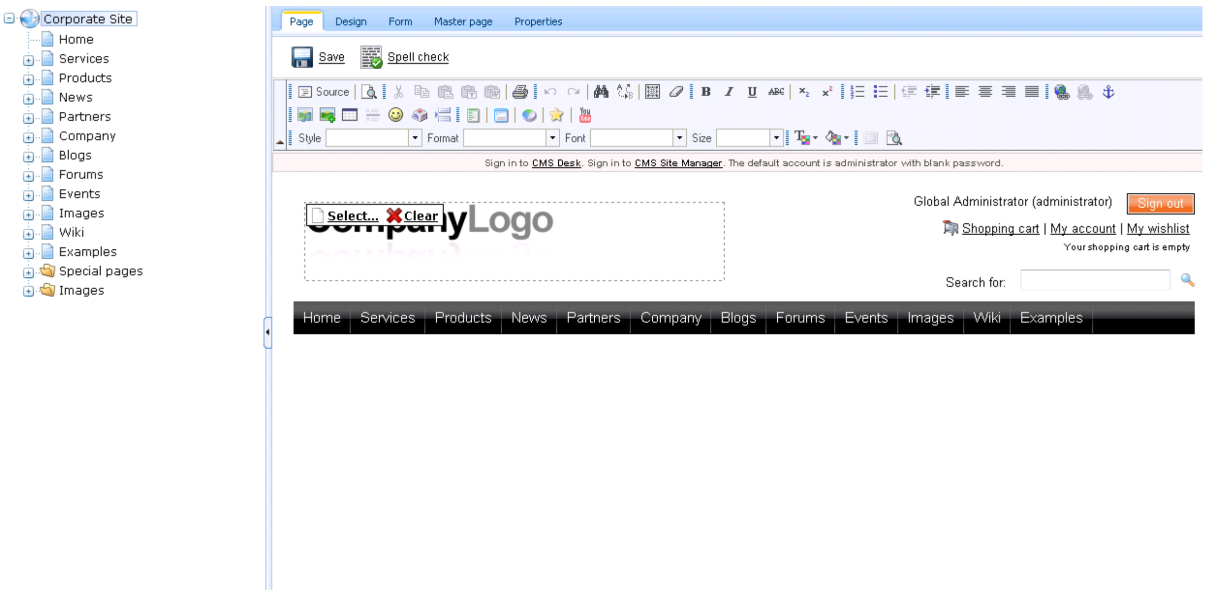
7.14.3 Changing logo of CMS Desk and CMS Site manager

You may decide to change a logo displayed at the top-left in CMS Desk and CMS Site manager.

1. Go to **<your KenticoCMS directory>\App_Themes\Default\Images\CMSDesk** for changing the logo in CMS Desk or to **<your KenticoCMS**

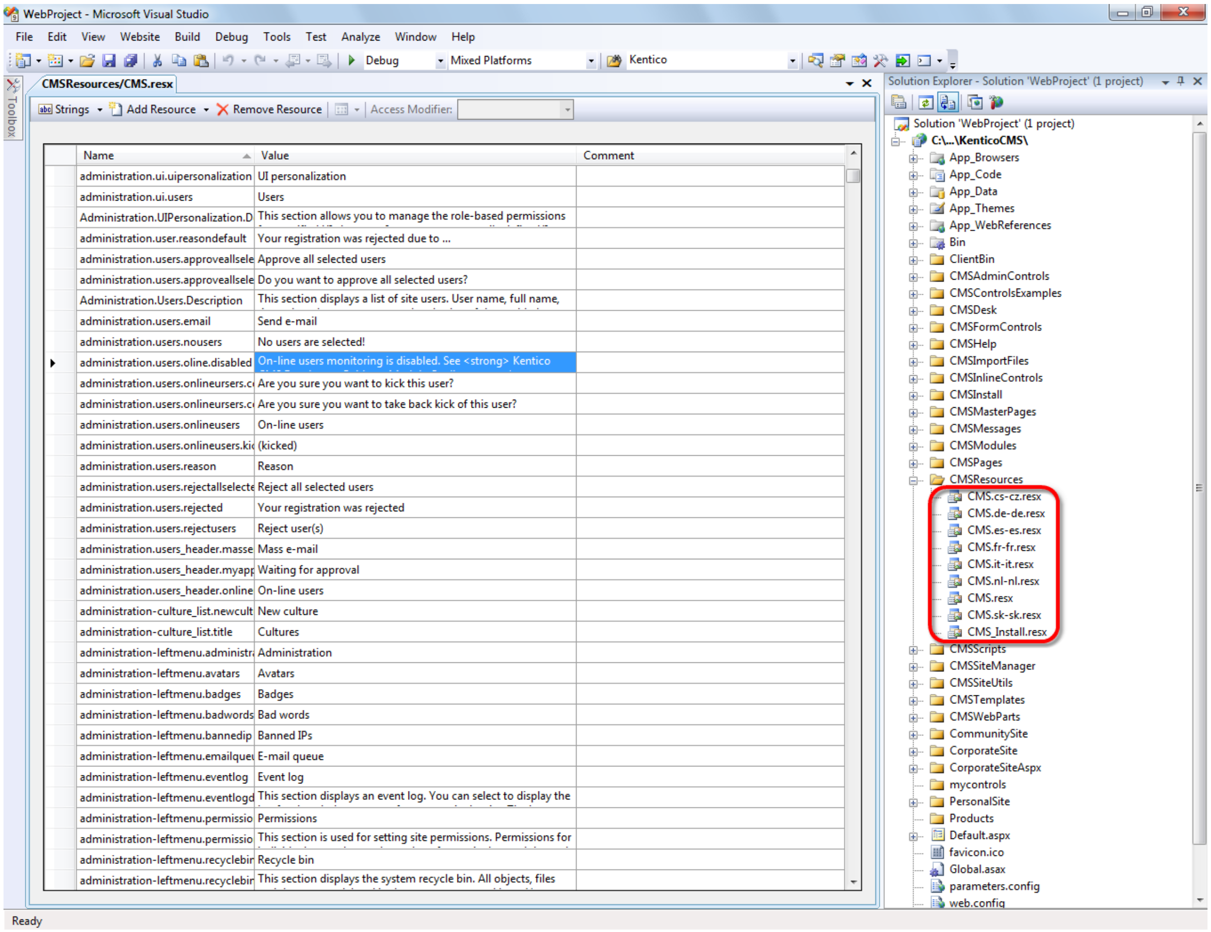
directory>\App_Themes\Default\Images\CMSSiteManager for changing the logo in CMS Site Manager.

2. In either directory, you can find **login.gif** file. All you have to do is to replace this file with your logo image. Please note that your image file has to be named **login.gif** as well.



7.14.4 Renaming resource strings

While re-branding Kentico CMS, you might need to check all the resource strings in the **.resx** files stored in the **~/CMSResources** folder and replace the occurrence of word **Kentico** with your own brand.



7.15 Scheduler

7.15.1 Overview

The Scheduler allows you to define when specified scheduled tasks should be executed. This is useful when you want some functionality to be performed automatically at a specific time or regularly over a certain time period.

In addition to other functions within the system, the scheduler is used to execute tasks for the following modules:

- Content staging
- E-commerce (E-commerce Guide -> Configuration settings -> Deleting old shopping carts)
- E-mail queue
- Newsletters
- On-line users
- Project management
- Smart search
- Time zones
- Web analytics
- Web farm synchronization

How it works

Individual tasks have a specified time interval that determines when they should be considered as ready to be executed. The exact time when tasks are actually executed depends on the settings of the scheduler.

By default, the scheduler performs a check at the end of standard page requests and runs the tasks that are ready to be executed. This means that tasks are executed only when user activity on your website is generating requests.

Alternatively, the scheduler can be configured to process tasks regularly according to an automatic internal timer, regardless of website activity. This can be done by adding the **CMSUseAutomaticScheduler** key into the **/configuration/appSettings** section of your **web.config** file, as shown below:

```
<add key="CMSUseAutomaticScheduler" value="true" />
```

The **Scheduler interval** setting found at **CMS Site Manager -> Settings -> System** determines how often the scheduler checks if there are any tasks ready to be executed. This setting can be useful even if tasks are checked based on page requests, since it sets the minimum interval between checks (e.g. to save resources on sites with high traffic). On the other hand, it is not recommended to set the value too high, since some parts of the system that utilize scheduled tasks work best when executed fairly often.

Additional low-level scheduler settings can be done by adding the keys listed in the Scheduler settings section of Appendix C - Web.config parameters.

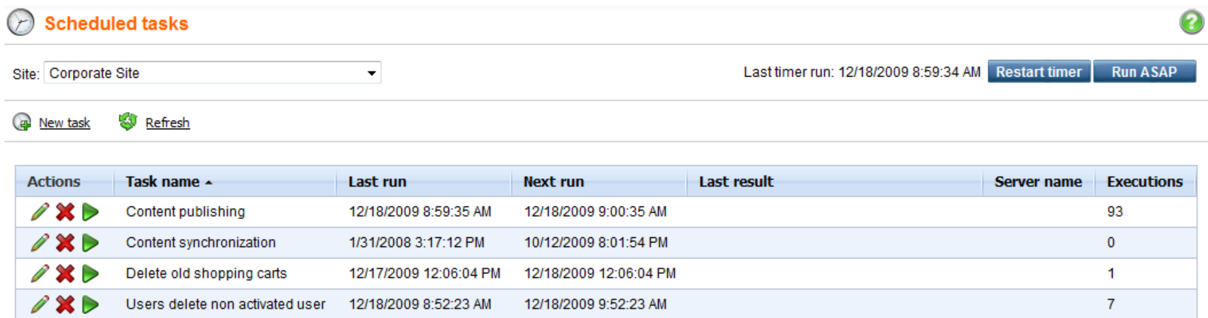
Scheduling reliability

Since the scheduling process runs within the ASP.NET process, tasks will not be executed if your website is not running. This happens when the process is recycled without being started again (after a long period of website inactivity).

If you want to run the scheduling reliably, it's necessary to ensure that your website is always running. You can do that by using some utility or an external service that requests the home page of your website on a regular basis.

Scheduled task administration

You can configure scheduled tasks in **Site Manager -> Administration -> Scheduled tasks**.



The **Site** drop-down list is used for selecting a site. After a site is selected, a list of tasks scheduled for the given site will be displayed in the table below. Tasks that are not site-specific can only be viewed when **(Global)** is selected. The tasks of the current site can also be edited in **CMS Desk -> Administration -> Scheduled tasks**.

Next to the site selection drop-down list, you can find the following two buttons:

- **Restart timer** - restarts the internal scheduling timer. This button is only available when the scheduler is configured to use the internal timer as described in the section above.
- **Run ASAP** - immediately executes all tasks that are ready to be executed.

The **New task** link allows you to schedule new custom tasks. The process of creating a scheduled task is covered in more detail in the Scheduling a custom code topic. The **Refresh** action updates the information displayed below in the list of tasks.

The three action icons on the left of each task serve the following purpose:

- **Edit** - allows the properties of the task (e.g. its execution interval) to be modified.
- **Delete** - removes the scheduled task.
- **Execute** - immediately runs the task. When a task is executed manually, the scheduled time of its next execution is automatically moved forward according to the scheduling interval.

The Scheduler internals and API sub-chapter provides information about the database tables and classes used by the module and examples of how scheduled tasks can be managed using the API.

7.15.2 Scheduling a custom code

The process of scheduling a custom task includes two steps: writing code to perform the required action and creating the scheduled task.

Writing task code

Task code must be placed into a specific class method.

1. Create a new assembly (Class Library) as a part of your solution and a new class inside this library. The following example uses assembly name **CMS.CustomTasks** and class **ShoppingCartCleaner**, however, you will need to use your own names.

2. References to the following dlls must be added to the newly created project. They can be found by using the **Browse** tab, located in the **Bin** folder of your CMS web project:

```
CMS.Scheduler.dll
CMS.Staging.dll
CMS.DataEngine.dll
CMS.SettingsProvider.dll
```

If your code uses any other dlls, references to them must be added here as well.

3. Add a reference to the new project to the **Bin** folder of your main CMS web project. It can be found under the **Projects** tab.

4. Define the **Execute(TaskInfo task)** method under the newly created class, which will be performed when the task is executed:

[C#]

```
...
using CMS.Scheduler;

namespace CMS.CustomTasks
{
    /// <summary>
    /// Provides an ITask interface to delete old shopping carts.
    /// </summary>
    public class ShoppingCartCleaner : ITask
    {
        public string Execute(TaskInfo task)
        {
            try
            {

                // The task code comes here

                return null;
            }
            catch (Exception ex)
            {
                return (ex.Message);
            }
        }
    }
}
```

5. **Build** the new project.

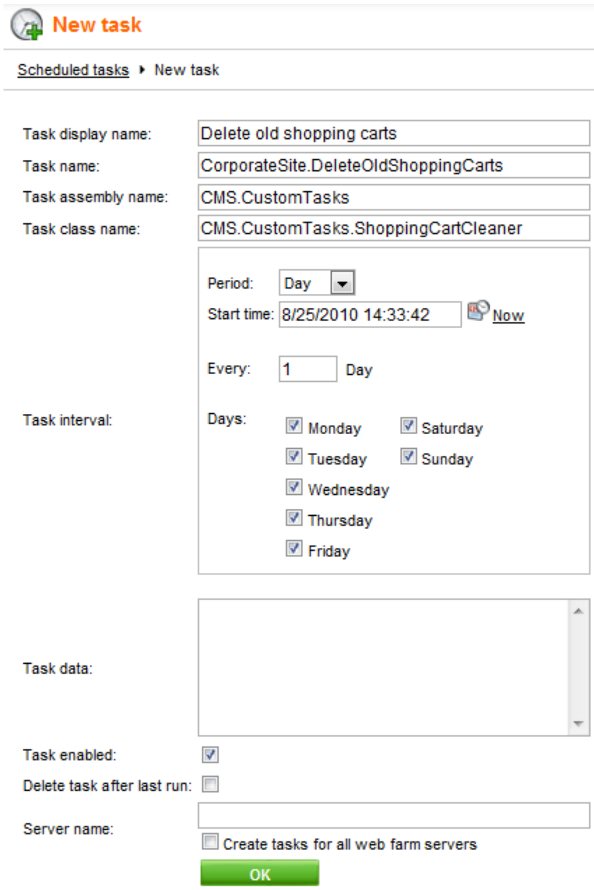
Creating a new task

1. Go to **Site Manager -> Administration -> Scheduled tasks.**
2. Select the site the task should be scheduled for.
3. Click **New task**.
4. Specify the task properties:

Task display name	Name used for the scheduled task shown in the administration

	interface.
Task name	Scheduled task name used in code.
Task assembly name	Name of the assembly. This example uses *CMS.CustomTasks*.
Task class name	Name of the class. This example uses *CMS.CustomTasks. ShoppingCartCleaner*.
Task interval	Sets the time interval between two executions of the task. This does not ensure that the task will be executed at the exact time, only that it will be considered ready to be executed. The precise execution time depends on the general settings of the scheduler.
Task data	Data which should be provided to the assembly. This field can be accessed from the code of the task, so it can be used as a parameter to easily modify the task without having to edit its implementation.
Task enabled	Indicates if the task is enabled.
Delete task after last run	Indicates if the task should be deleted after its final run (applicable if the task is set to run only once).
Server name	Name of the server where the task is run.
Create tasks for all web farm servers	If checked, tasks will be created for all web farm servers and the **Server name** field will be grayed out.

5. Click **OK**.
6. The task is now scheduled.



7.15.3 Scheduler internals and API

7.15.3.1 Database tables and API classes

The following database table is used by the scheduler:

- **CMS_ScheduledTask** - contains scheduled tasks and their settings

The scheduler API is provided by the following **CMS.Scheduler** namespace classes:

- **SchedulingExecutor** - this class provides methods for executing scheduled tasks
- **SchedulingHelper** - this class provides functionality for managing the scheduler
- **TaskInfo**, **TaskInfoProvider** - these classes provide functionality for managing scheduled tasks
- **TaskInterval** - this class provides functionality for managing the time intervals used by scheduled tasks

The following topics show examples of how these classes can be used:

- Getting scheduled task data
- Managing scheduled tasks
- Running a scheduled task

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

7.15.3.2 Getting scheduled task data

The following sample code shows how you can get a scheduled task as a *TaskInfo* object:

[C#]

```
using CMS.Scheduler;
using CMS.CMSHelper;

...

        int taskID = 1;
        int siteID = CMSContext.CurrentSiteID;
        string taskName = "Content.Publish";
        string siteName = "CorporateSiteASPX";

        // Get object
        TaskInfo ti = TaskInfoProvider.GetTaskInfo(taskID);

        TaskInfo ti2 = TaskInfoProvider.GetTaskInfo(taskName, siteID);

        TaskInfo ti3 = TaskInfoProvider.GetTaskInfo(taskName,siteName);
```

The following sample code shows how you can get a DataSet containing all scheduled tasks in the system:

[C#]

```
using System.Data;
using CMS.Scheduler;

...

        // Get DataSet of all scheduled tasks
        DataSet ds = TaskInfoProvider.GetAllTasks();
```

7.15.3.3 Managing scheduled tasks

The following sample code shows how you can modify the *Display name* of an existing scheduled task:

[C#]

```
using CMS.Scheduler;
```

```
...

        string taskName = "TestContentPublishing";
        string siteName = "CorporateSiteASPX";

        // Get task object
        TaskInfo ti = TaskInfoProvider.GetTaskInfo(taskName, siteName);

        // If task exists
        if (ti != null)
        {
            // Update property
            ti.TaskDisplayName += " updated";

            // Save to database
            TaskInfoProvider.SetTaskInfo(ti);
        }
```

The following sample code shows how you can create a new scheduled task and add it to the system:

[C#]

```
using System;
using System.Collections;
using CMS.Scheduler;
using CMS.CMSHelper;

...

        // Create new task object
        TaskInfo ti = new TaskInfo();

        // Set properties
        ti.TaskAssemblyName = "CMS.WorkflowEngine";
        ti.TaskClass = "CMS.WorkflowEngine.ContentPublisher";
        ti.TaskDisplayName = "Test content publishing";
        ti.TaskName = "TestContentPublishing";

        // Create interval
        TaskInterval interval = new TaskInterval();

        // Set interval properties
        interval.Period = SchedulingHelper.PERIOD_DAY;
        interval.StartTime = DateTime.Now;
        interval.Every = 2;

        // Add some days to interval
        ArrayList days = new ArrayList();
        days.Add(DayOfWeek.Monday.ToString());
        days.Add(DayOfWeek.Sunday.ToString());
        days.Add(DayOfWeek.Thursday.ToString());
```

```
        interval.Days = days;

        ti.TaskInterval = SchedulingHelper.EncodeInterval(interval);
        ti.TaskSiteID = CMSContext.CurrentSiteID;
        ti.TaskData = "<data></data>";
        ti.TaskEnabled = true;
        ti.TaskNextRunTime = SchedulingHelper.GetNextTime(ti.TaskInterval, DateTime.

        // Save to database
        TaskInfoProvider.SetTaskInfo(ti);
```

The following sample code shows how you can delete an existing scheduled task:

[C#]

```
using CMS.Scheduler;

...

        string taskName = "TestContentPublishing";
        string siteName = "CorporateSiteASPX";

        // Get task object
        TaskInfo ti = TaskInfoProvider.GetTaskInfo(taskName, siteName);

        // If task exists
        if (ti != null)
        {
            // Delete from database
            TaskInfoProvider.DeleteTaskInfo(ti);
        }
```

7.15.3.4 Running a scheduled task

The following sample code shows how you can run an existing scheduled task:

[C#]

```
using CMS.Scheduler;

...

        string taskName = "TestContentPublishing";
        string siteName = "CorporateSiteASPX";

        // Get task object
        TaskInfo ti = TaskInfoProvider.GetTaskInfo(taskName, siteName);

        // If task exists
```

```
        if (ti != null)
        {
            // Run task
            SchedulingExecutor.ExecuteTask(ti);

        }
```

7.16 SQL import

7.16.1 How to use the SQL import utility

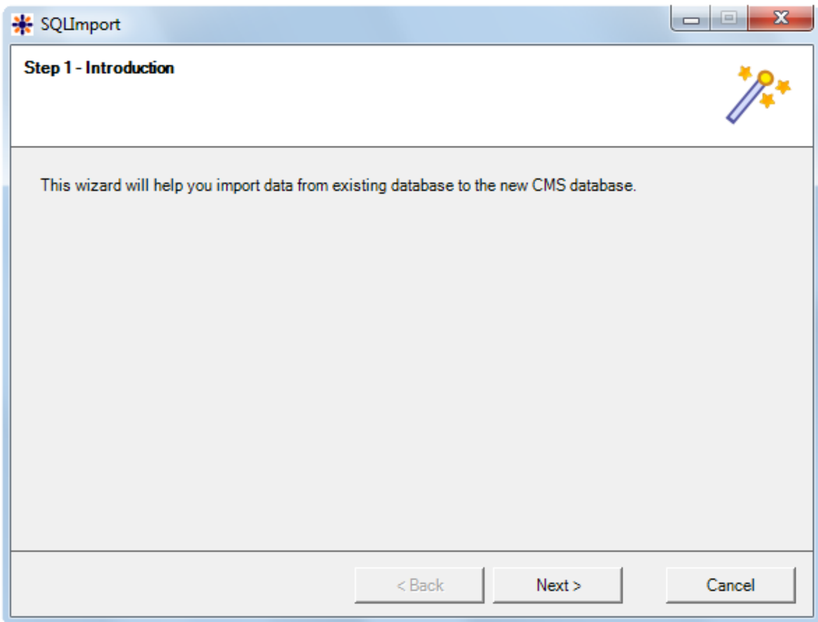
The SQL Import utility allows you to import data from a Microsoft SQL Server database as new documents into the Kentico CMS content repository (content tree). If your source data is in a different format, you may need to import them to some Microsoft SQL Server database using Microsoft SQL Server tools.

Before you run the SQL Import utility, you need to configure its parameters. The utility is stored in folder *<Kentico CMS installation in Program Files>\SQLImport*. Open the **SQLImport.exe.config** file and set the following values:

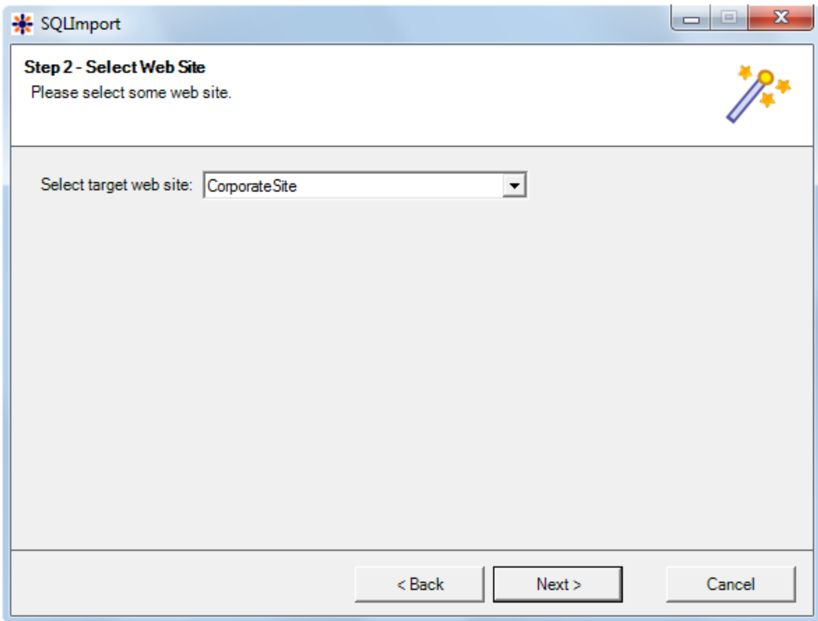
Key	Description	Sample Value
SourceConnectionString	Connection string to the source database.	`<add name=" SourceConnectionString" connectionString="Persist Security Info=False; database=mysourcedb; server=myserver;user id=sa; password=mypassword;Current Language=English;Connection Timeout=120;" />`
CMSConnectionString	Connection string to the Kentico CMS database. You can copy this value from the web.config file of your web project.	`<add name=" CMSConnectionString" connectionString="Persist Security Info=False; database=mycmsdb; server=myserver;user id=sa; password=mypassword;Current Language=English;Connection Timeout=120;" />`
CMSDefaultUserID	*UserID* value of the user who will be author of the imported files. The default value is 53 for the administrator user account. You can find other values in the *CMS_User* table.	`<add key="CMSDefaultUserID" value="53" />`
ImportFilesDiskPath	The physical disk path to the folder	`<add key="ImportFilesDiskPath`

	with files to be imported (in case you are importing files together with data).	" value="c:\Temp" />
WebApplicationPhysicalPath	Path to the root of the web project folder (in case the target site is set to save files to disk [**Site Manager -> Settings -> File -> Store files in filesystem**]).	<add key=" WebApplicationPhysicalPath" value="C:\Website" />

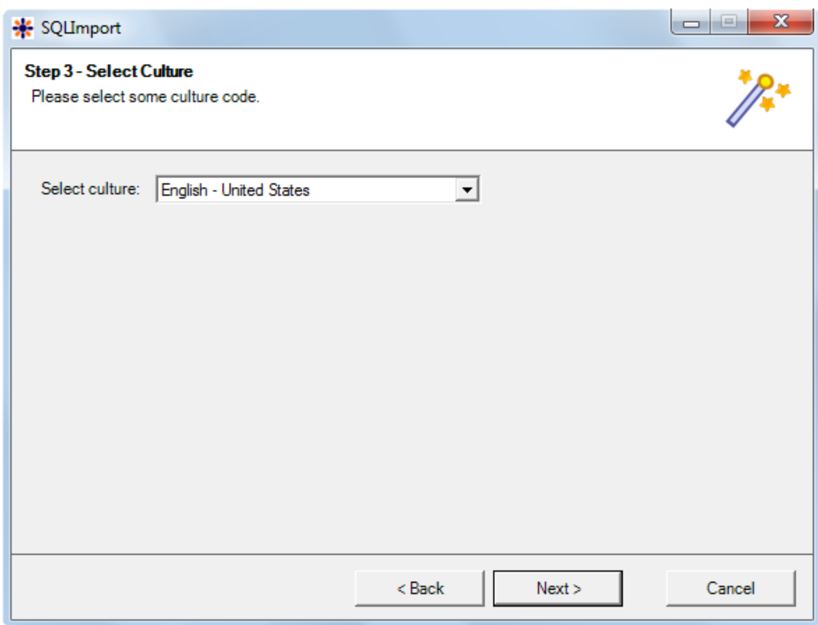
Now you can start the import utility using **SQLImport.exe**. The import wizard starts:



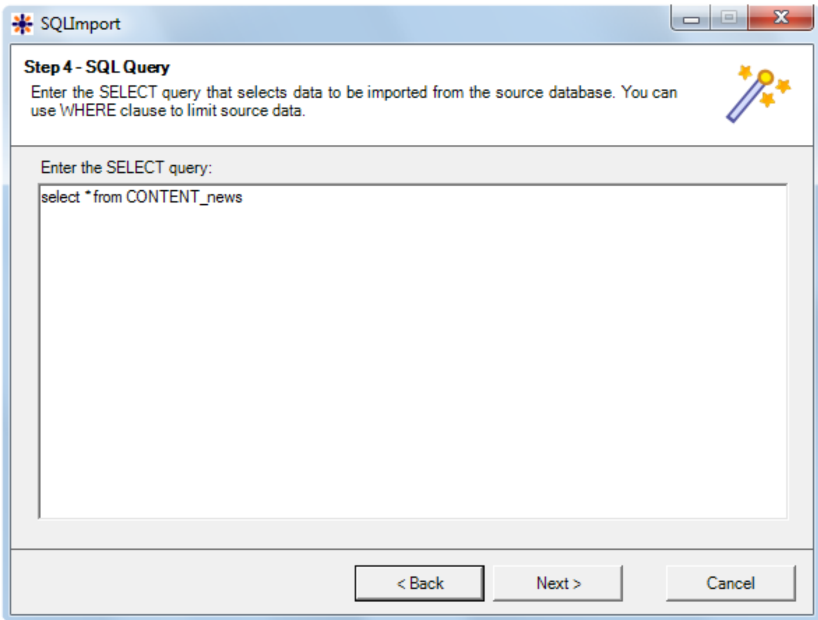
Click **Next**. Select the target website:



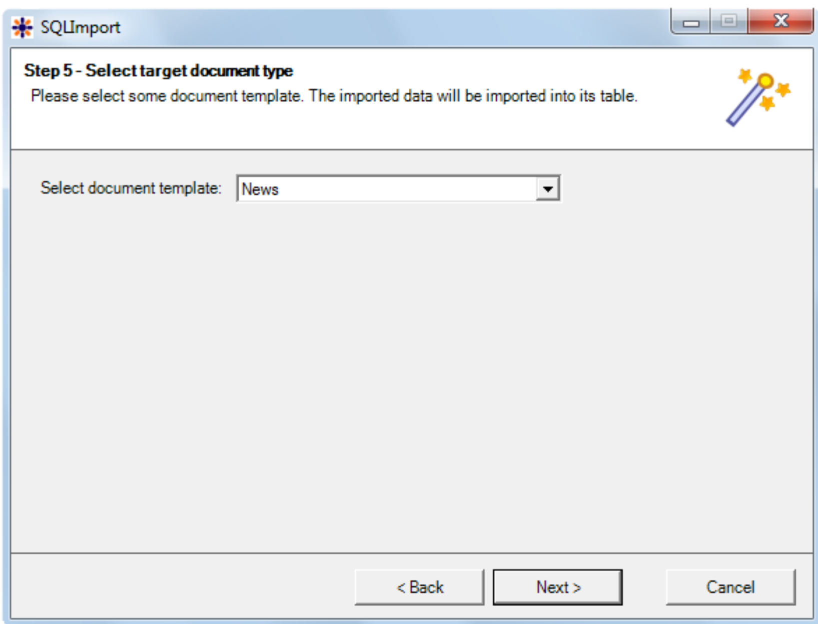
Click **Next**. Select the target content culture:



Click **Next**. Enter the SQL query that will retrieve the data to be imported from the source database:

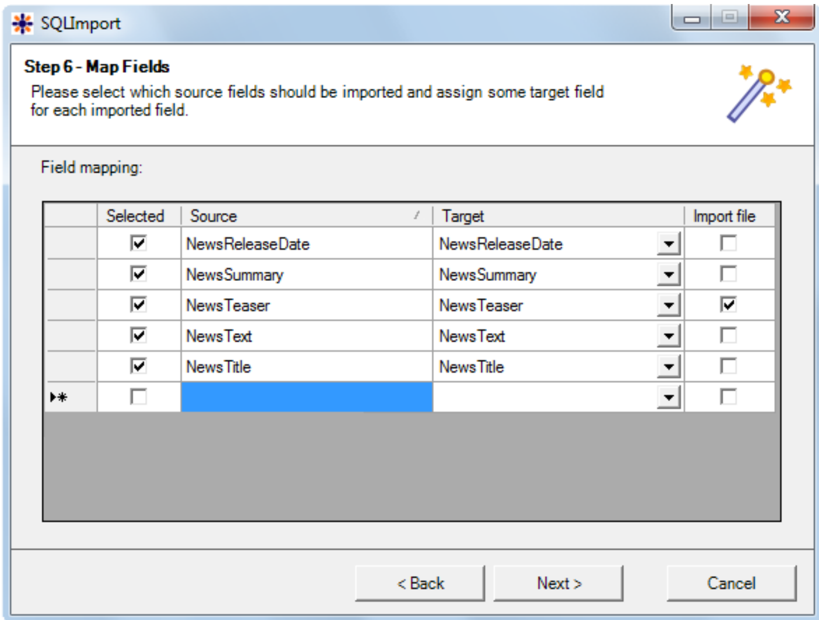


Click **Next**. Select the type of the imported documents.

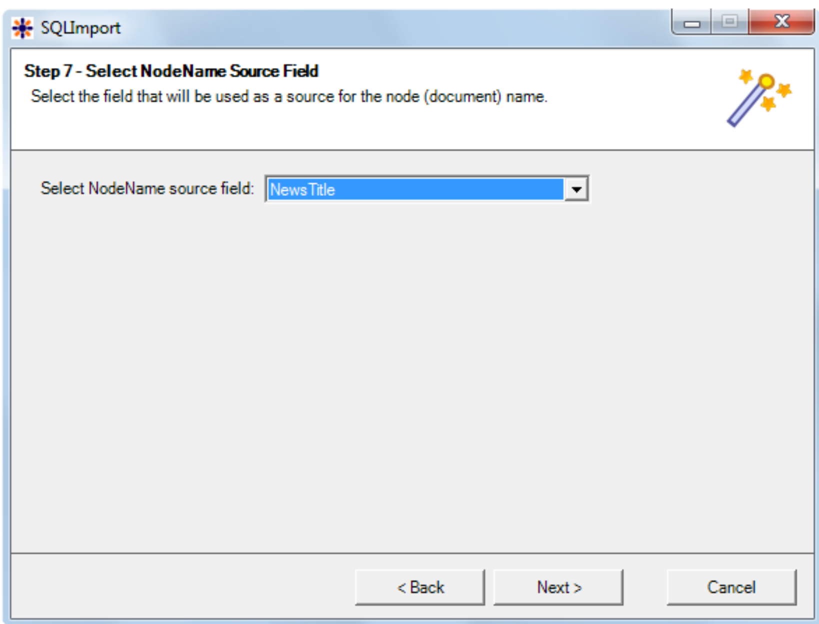


Click **Next**. Map the source and target fields and check the fields that should be imported. You can add the source fields manually multiple times if you need to import one source field into several target fields.

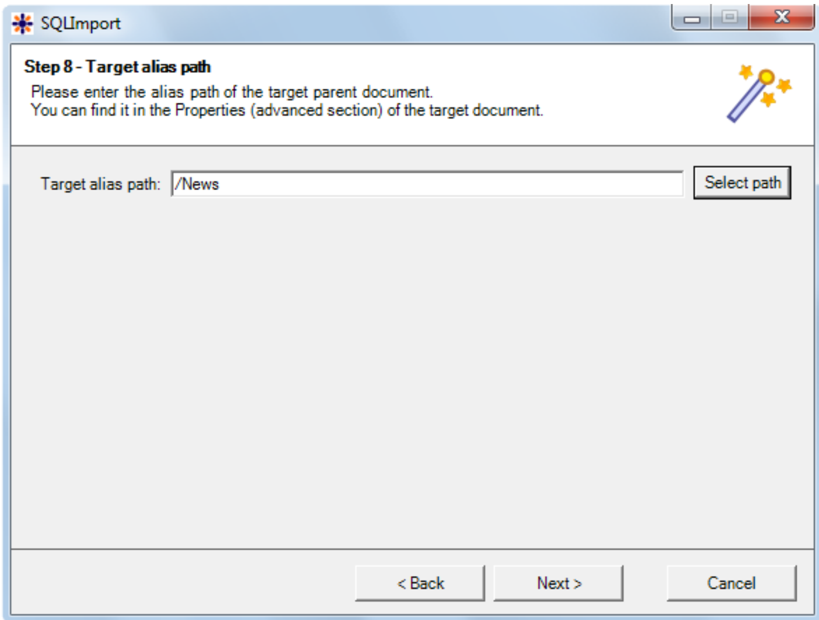
If the source field contains the path to the file, check the **Import file** box. In this case, the source field must contain a relative path to the path in the *ImportFilesDiskPath* web.config parameter - both values are joined and the file is imported from the specified location.



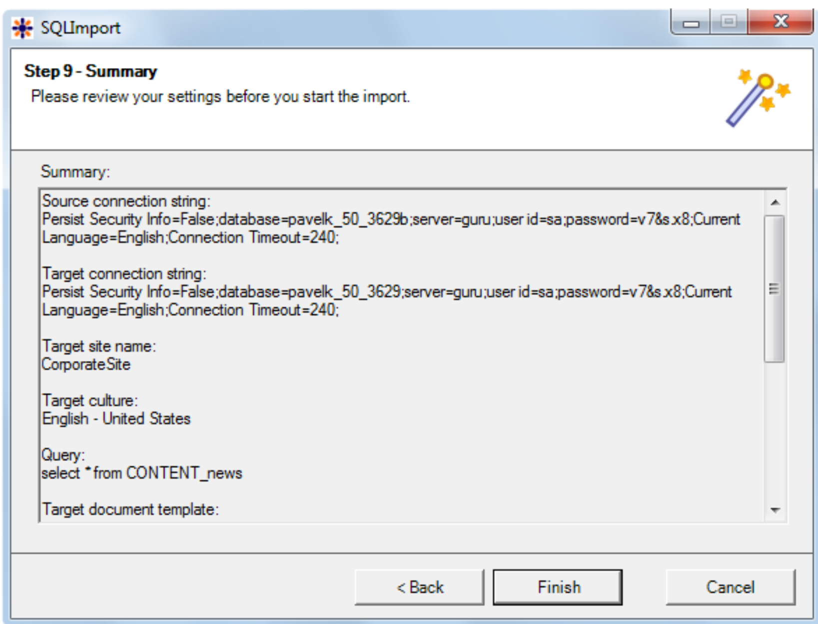
Click **Next**. Select the field that will be used for the name of the document:



Click **Next**. Select the target alias path where the documents should be imported:



Click **Next**. Review the import settings:



Now the data is imported into the Kentico CMS database.

7.17 System tables and custom fields

7.17.1 Overview

Kentico CMS allows you to modify some of the system tables and enhance them with custom attributes. You can edit them in **Site Manager -> Development -> System tables**.

When you add a new column to the user profile, it is available on the **Custom fields** tab of the **User**

properties dialog.

> **Always allow empty values**
>
> When creating a new empty field in a system table, it is highly recommended that you always allow empty values in the newly created field, so that the system procedures and existing dialogs keep working.

- If you would like to learn how to add custom data to all documents, please refer to the Custom document data topic.

7.17.2 Custom document data

In some cases, you may need to add custom data to all documents. In this case, you can use the *NodeCustomData* or *DocumentCustomData* (culture specific) fields in the *CMS_Tree* and *CMS_Document* database tables, respectively.

These fields are accessible through the following properties of the document (TreeNode):

- *TreeNode.NodeCustomData*
- *TreeNode.DocumentCustomData*

You can use these values in two ways:

1. You can use them as a single ntext block of text:

[C#]

```
TreeNode.NodeCustomData.Value = "my value";
```

2. You can use them as a collection of custom values that are stored as an XML document:

[C#]

```
TreeNode.NodeCustomData["myproperty1"] = "my value 1";
TreeNode.NodeCustomData["myproperty2"] = "my value 2";
```

7.18 Web parts

7.18.1 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 (**Design website** from the **Modules-> Content** permission matrix) 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.

From a developer's point of view, a web part is a user control (ASCX) that inherits from the **CMSAbstractWebPart** class. You can easily create your own web parts as described in the Developing web parts topic.

The Kentico CMS installation contains many built-in web parts, but sometimes it may be necessary to modify the behaviour, design or functionality of one of them. The Modifying web parts sub-chapter covers several ways how this can be done.

All web parts have the option of using an AJAX UpdatePanel. More information can be found in the AJAX support topic.

Also of interest may be Widgets, which provide functionality similar to web parts, but allow further personalization by users.

Web part management

Web parts can be managed in **Site Manager -> Development -> Web parts**. Here you can see all web parts organized into categories in a tree structure. Each web part has the following properties on the **General** tab:

- **Display name** - name of the web part displayed to the users in the administration interface
- **Code name** - name of the web part used in website code
- **Category** - here you can choose the category of the web part catalog where the web part will be placed
- **Type** - type of the web part; the following web part types are available:
  - **Standard** - typical web parts displaying some content
  - **Data source** - do not display any content, only provide data to be displayed by a connected web part
  - **Filter** - can be connected to a data source and enables users to limit the data provided by it
  - **Placeholder** - used for the **General -> Layout -> Page placeholder** web part; specifies the 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
- **File name** - contains a relative path to the user control that implements the web part; the path starts from the CMSWebParts folder; it's recommended that you organize the web parts on the disk in the same way as in the categories; example: *Search/cmscompletesearchdialog.ascx*
- **Description** - text describing the web part, it will be displayed in the web part catalog
- **Thumbnail** - image used in the web part catalog

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

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

Default

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

Visibility

- **Visible -** indicates if the web part should be displayed
- **Hide on subpages -** indicates if the web part should be hidden on sub-pages
- **Show for document types -** the list of document type code names separated with a semicolon (;) for which the web part should be displayed (if they are selected as the current document)
- **Display to roles -** the list of roles the web part should be displayed to

Web part container

- **Web part container -** container (box) displayed around the web part; this value contains the name of the container defined in *Site Manager -> Development -> Web part containers*
- **Container title -** container title displayed if you specify the container; displayed only if the *{% ContainerTitle%}* macro is used in the code of the container
- **Container CSS class -** CSS class used for the web part container; applied only if the *{% ContainerCSSClass%}* macro is used as a value of the Class attribute in the code of the container
- **Container custom content -** custom content of the web part container; applied only if the *{% ContainerCustomContent%}* macro is used in the code of the container

HTML Envelope

- **Content before -** HTML content displayed before the web part; you can use it to display a header or add some encapsulating code, such as *<div>* or *<table>* elements for the required layout
- **Content after -** HTML content displayed after the web part; you can use it to display a footer or close the tags contained in the *ContentBefore* value, such as *</div>* or *</table>* elements

AJAX

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

Time zones

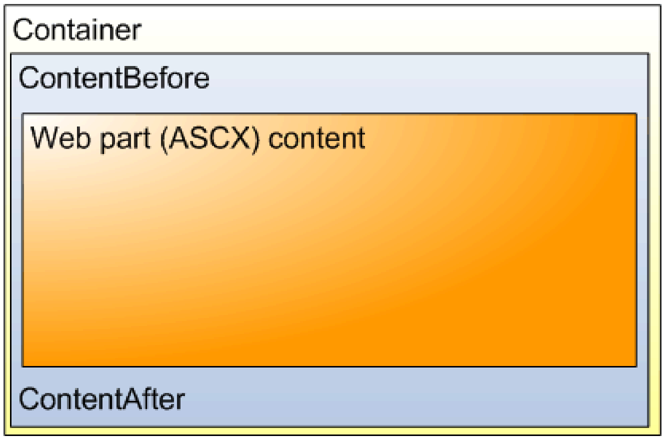
- **Time zone -** time zone used for the web part content
- **Custom time zone -** optional custom time zone that can be used specifically only for this web part

Partial caching

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

specific web part.

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



The containers, unlike the ContentBefore and ContentAfter sections, are re-usable and they can contain dynamically inserted values of web part properties.

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

Web part documentation

You can add your documentation to the web part on the **Documentation** tab. If you wish to document particular properties, you need to fill in the **Field description** on the **Properties** tab.

You can generate complete web part documentation and print it by going to *<website URL>/CMSPages/ Dialogs/documentation.aspx?generate=full* in your browser. If you want to print it, it's recommended that you use FireFox for correct formatting and page breaking.

7.18.2 Developing web parts

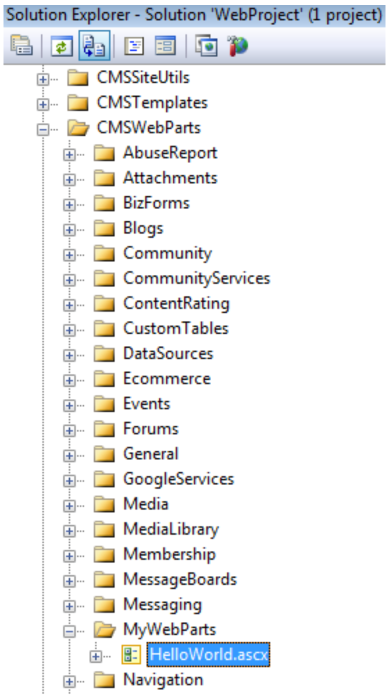
This chapter will guide you through the process of creating a very simple "Hello world" web part that displays a label and a button. When the button is clicked, it updates the current time displayed by the label.

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.

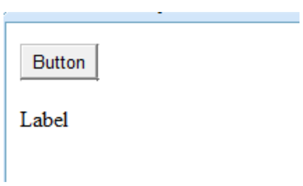
2. Right-click the **CMSWebParts** folder in the **Solution Explorer** window and choose **New Folder**.

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 onto 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)this.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(My.GetValue("ButtonText"), String)
End Sub
```

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

9. Save all changes.

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

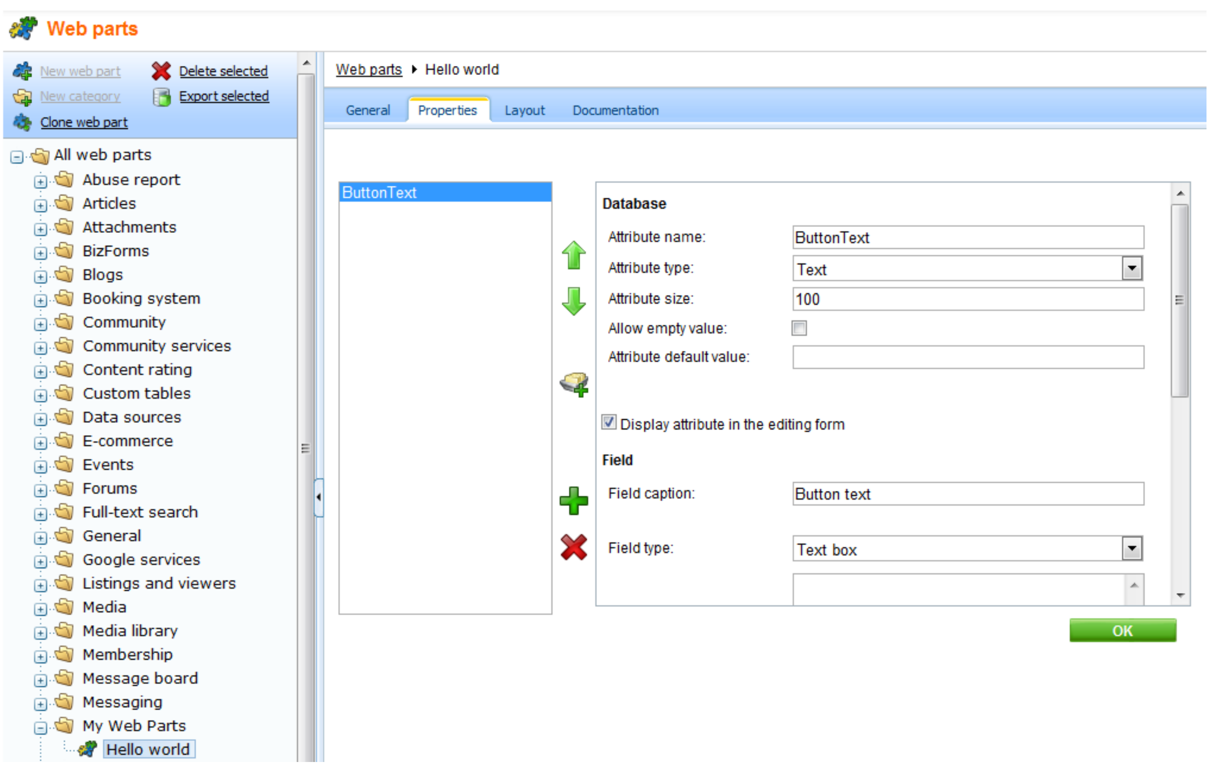
11. Click the new category and click **New web part**. Choose to **create a new web part** and enter the following values:

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

Click **OK**.

12. Switch to the **Properties** tab and add () the following property:

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

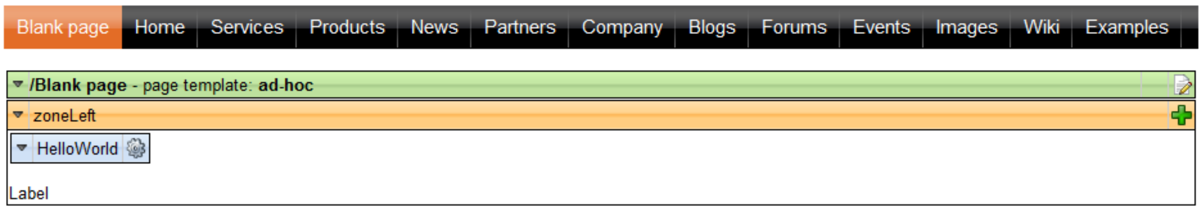


Click **OK**.

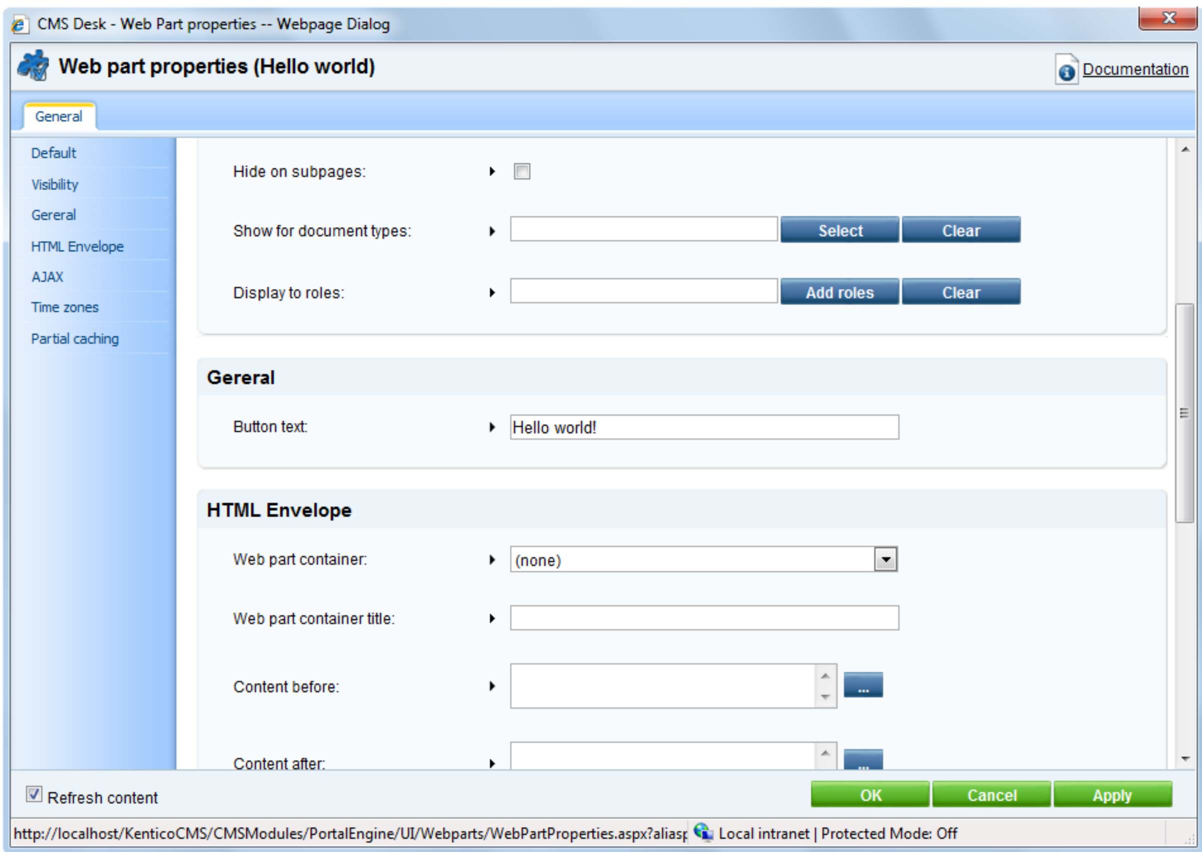
13. Switch to **CMS Desk**.

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

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



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



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

You have learned how to create a simple web part.

Tip: Displaying content on the web part

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

Error message "The control collection cannot be modified during DataBind, Init, Load, PreRender or Unload phases."

If you get this error message you may need to modify the code of your web part, so that it doesn't display any content on the Design tab - for example:

[C#]

```
using CMS.PortalEngine;

public override void OnContentLoaded()
{
  base.OnContentLoaded();
  if ((this.PagePlaceholder.ViewMode == ViewModeEnum.Design)
     || (this  .HideOnCurrentPage) || (!this.IsVisible))
    {
         this.Repeater1.DataSourceID = "";
         this.CMSRepeater1.StopProcessing = true;
    }
}
```

Initializing Kentico CMS controls in your custom web parts

If you are using Kentico CMS controls in your web parts, it is recommended to initialize the controls' properties using a combination of the **OnContentLoaded** and **SetupControl** methods. This is the way it is handled in all Kentico CMS web parts. You can view the code of any of the web parts located in *<project folder>/CMSWebParts* and use it as an example.

If you are using classic .NET controls or third party controls, this can be handled in the **PageLoad** method.

> If a problem occurs (e.g. on postback), try loading the control dynamically. This can be achieved by making a control from the web part and loading it dynamically (e.g. using the **General -> User control** web part).

7.18.3 Modifying web parts

7.18.3.1 Modifying web parts

If you need to modify the behavior of a standard web part, you have the following options according to your requirements:

1. **You only need to set the web part properties dynamically in your code**

You can create a user control, add the given web part to it and write additional code. See the Setting web part properties dynamically in your code topic.

2. **You need to modify the design (layout) of a web part**

You can use the custom web part layouts described in the Customizing web part layout topic.

3. **You need to modify the code of a web part**

You need to create a copy of a standard web part as described in the Modifying the code or design of standard web parts topic.

4. **You need to create a specialized version of a web part with different default property values**

You can use Web part inheritance to easily create a derived web part.

7.18.3.2 Setting web part properties dynamically in your code

In some cases, you may need to set the values of web part properties in your code, depending on some particular business rules. In such case, you need to create a new ASCX user control and place the original web part onto this user control. In the user control code, you can implement your custom logic and set the properties appropriately.

Example:

The following example shows how you can dynamically set the **WHERE condition (WhereCondition)** property of the **Repeater** web part based on if the current user is or is not authenticated. It uses the standard *News* document type with a custom boolean type field: **Show to public users (ShowToPublicUsers)**.

1. Open the web project in **Visual Studio**.

2. Create a **New folder** under the project root called *CMSGlobalFiles* (if it doesn't already exist). This location will ensure that your user controls can be exported along with the site when it is deployed to the live server.

3. Create a new **Web User Control** under the **CMSGlobalFiles** folder and name it **NewsRepeater.**

ascx.

4. Switch to the **Design** tab and drag and drop **CMSWebParts/Viewers/Documents/cmsrepeater.ascx** from the Solution Explorer onto your user control. You could alternatively use the CMSRepeater server control, but this is not the purpose of this example. Set its properties like this:

- **ID:** RepeaterWebPart1
- **ClassNames:** cms.news (document types)
- **Path:** /news/%
- **TransformationName:** cms.news.preview
- **SelectedItemTransformationName:** cms.news.default

5. Now add the following to the code behind of your user control inside the **CMSGlobalFiles_NewsRepeater** class:

[C#]

```
protected void Page_Init(object sender, EventArgs e)
{
    if (CMS.CMSHelper.CMSContext.CurrentUser.IsPublic())
    {
        // public user - show only public news
        this.RepeaterWebPart1.WhereCondition = "ShowToPublicUsers = 1";
        this.RepeaterWebPart1.ReloadData();
    }
}
```

This will set the **WhereCondition** property value dynamically depending on whether the current user is signed in. **Save** all changes.

6. Go to **Site Manager -> Development -> Document types -> News**, add a **New attribute** (➕) on the **Fields** tab and set its properties as shown below:

- **Attribute name**: ShowToPublicUsers
- **Attribute type**: Boolean (Yes/No)
- **Field caption**: Show to public users
- **Field type**: Check box

7. Go to **CMS Desk -> Content**, choose **Home**, switch to the **Design** tab and add (➕) a new **General/User control** web part to the **zoneCenter** zone. Set the **User control virtual path** property value to: *~/CMSGlobablFiles/NewsRepeater.ascx*

8. Edit some news document in the **/News** section of the website on the **Form** tab and check the **Show to public users** checkbox.

9. **Sign out** and view the home page. You should see only news items that you marked as **Show to public users**.

You have learned how to dynamically set web part properties based on your custom logic.

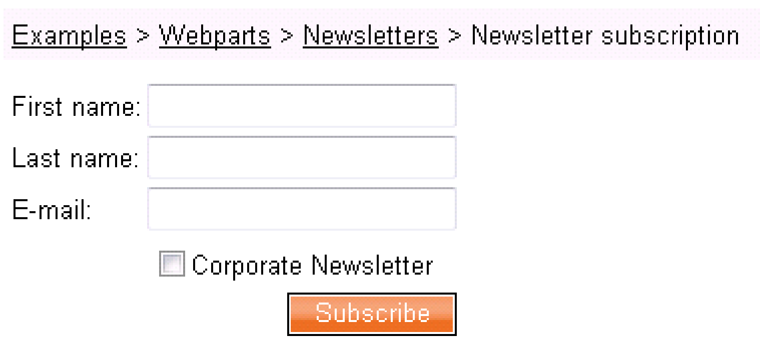
7.18.3.3 Customizing web part layout

Kentico CMS comes with many built-in web parts and their appearance and design can easily be modified to fit your specific requirements. The concept of web part layouts allows you to customize the look of a web part by defining a custom HTML layout. So the web part layout is basically a custom skin for the web part.

The layout of a specific web part can be selected by configuring ( ) it on the **Design** tab of **CMS Desk** and switching to the **Layout** tab. Here you can select from a list of layouts created for the current web part, or write a new layout. This list of layouts can also be managed at **CMS Site Manager -> Development -> Web parts -> .. select the given web part from the tree ... -> Layout**.

Example: Customizing Newsletter subscription dialog

In this example, we will customize the newsletter subscription dialog layout. The standard layout looks like this (when no user is logged on):



Go to **CMS Desk -> Content** and navigate to the **Examples -> Web parts -> Newsletters -> Newsletter subscription** page (if you're using the sample Corporate Site).

Switch to the **Design** tab and configure ( ) the **Newsletter subscription** web part. Click the **Layout** tab and choose *(New)* from the drop-down list. Enter the following values:

- **Display name:** Narrow layout
- **Code name:** NarrowLayout

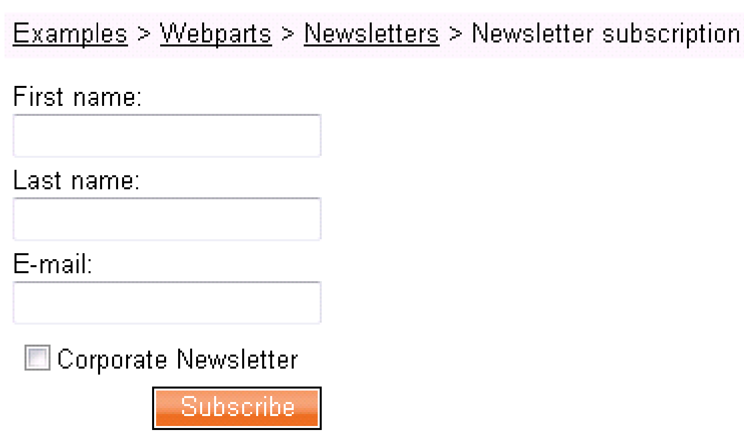
Enter the following HTML code:

```
<%@ Control Language="C#" AutoEventWireup="true" Inherits
="CMSWebParts_Newsletters_NewsletterSubscriptionWebPart" CodeFile="~/CMSWebParts/
Newsletters/NewsletterSubscriptionWebPart.ascx.cs" %>

<%@ Register Src="~/CMSFormControls/Inputs/SecurityCode.ascx" TagName
="SecurityCode" TagPrefix="cms" %>

<asp:Panel ID="pnlSubscription" runat="server" DefaultButton="btnSubmit" CssClass
="Subscription">
    <asp:Label runat="server" ID="lblInfo" CssClass="InfoMessage" EnableViewState
="false"
        Visible="false" />
    <asp:Label runat="server" ID="lblError" CssClass="ErrorMessage"
```

```
EnableViewState="false"
        Visible="false" />
    <div class="NewsletterSubscription">
        <table cellspacing="0" cellpadding="0" border="0" class="Table">
            <asp:PlaceHolder runat="server" ID="plcFirstName">
                <tr>
                    <td>
                        <cms:LocalizedLabel ID="lblFirstName" runat="server"
AssociatedControlID="txtFirstName"
                            EnableViewState="false" />
                    <br />
                        <asp:TextBox ID="txtFirstName" runat="server" CssClass
="SubscriptionTextbox"
                            MaxLength="200" />
                    </td>
                </tr>
            </asp:PlaceHolder>
            <asp:PlaceHolder runat="server" ID="plcLastName">
                <tr>
                    <td>
                        <cms:LocalizedLabel ID="lblLastName" runat="server"
AssociatedControlID="txtLastName"
                            EnableViewState="false" />
                    <br />
                        <asp:TextBox ID="txtLastName" runat="server" CssClass
="SubscriptionTextbox"
                            MaxLength="200" />
                    </td>
                </tr>
            </asp:PlaceHolder>
            <asp:PlaceHolder runat="server" ID="plcEmail">
                <tr>
                    <td>
                        <cms:LocalizedLabel ID="lblEmail" runat="server"
AssociatedControlID="txtEmail"
                            EnableViewState="false" />
                    <br />
                        <asp:TextBox ID="txtEmail" runat="server" CssClass
="SubscriptionTextbox"
                            MaxLength="400" />
                    </td>
                </tr>
            </asp:PlaceHolder>
            <asp:PlaceHolder runat="server" ID="plcNwsList">
                <tr>
                    <td>
                        <asp:CheckBoxList runat="server" ID="chklNewsletters"
CssClass="NewsletterList" />
                    </td>
                </tr>
            </asp:PlaceHolder>
            <asp:PlaceHolder runat="server" ID="plcCaptcha">
                <tr>
                    <td>
                        <cms:LocalizedLabel ID="lblCaptcha" runat="server"
AssociatedControlID="scCaptcha"
                            EnableViewState="false" />
                    <br />
                        <cms:SecurityCode ID="scCaptcha" GenerateNumberEveryTime
```

```
="false" runat="server" />
                    </td>
                </tr>
            </asp:PlaceHolder>
            <asp:PlaceHolder ID="pnlButtonSubmit" runat="server">
                <tr>
                    <td align="right">
                        <cms:LocalizedButton ID="btnSubmit" runat="server" OnClick
="btnSubmit_Click"
                            CssClass="SubscriptionButton" EnableViewState="false"
/>
                    </td>
                </tr>
            </asp:PlaceHolder>
            <asp:PlaceHolder ID="pnlImageSubmit" runat="server">
                <tr>
                    <td align="right">
                        <asp:ImageButton ID="btnImageSubmit" runat="server"
OnClick="btnSubmit_Click"
                            EnableViewState="false" />
                    </td>
                </tr>
            </asp:PlaceHolder>
        </table>
    </div>
</asp:Panel>
```

Click **OK**. When you look at the page now, you will see that the dialog looks like this (when no user is logged on):



Do not remove any controls from the layout

It's important to **keep all the controls** in the layout. If you need to hide some of them, you can add the *Visible="False"* attribute to the control, but the control must stay in the layout to allow the web part to keep working.

This issue **may also cause problems when upgrading to a new Kentico CMS version** - if some of the built-in web parts use a new control and you use your web part layout created in the previous version, the web part may stop working. Please be sure to test your website after an upgrade carefully if you're using web part layouts.

7.18.3.4 Modifying code of standard web parts

This topic explains how you can create a copy of a standard web part and modify its code.

The following example shows how you can send a custom e-mail when a BizForm form is submitted and display a custom confirmation message. It uses the *CMSWebParts\BizForms\BizForm.ascx* web part as its base and adds a custom handler to the *OnAfterSave* event.

1. Create a copy of the BizForm web part in Kentico CMS. Go to **Site Manager -> Development -> Web parts -> BizForms -> BizForm**. Click **Clone web part** and enter the following values:

- **Display name:** BizForm with custom e-mail
- **Code name:** BizFormWithEmail
- **Category:** BizForms
- **File name:** BizForms/bizformwithemail.ascx
- **Clone web part files:** yes (checked)

Click **Clone**. The system creates a copy of the existing web part using the new name and it also copies the code (ASCX and CS file).

2. Now we will make the modifications to the web part. Open the web project using the **WebProject.sln** file in Visual Studio and edit the **CMSWebParts/BizForms/bizformwithemail.ascx** file.

3. Switch to the **Design** tab and drag and drop a **Label** control onto the page. Set its **ID** to *lblConfirmationMessage* and clear its **Text** property.

4. Click the BizForm control and choose **Events** in the **Properties** window. Double-click the **OnAfterSave** event, which will create the **viewBiz_OnAfterSave()** method, and modify it to look like this:

[C#]

```
protected void viewBiz_OnAfterSave()
{
    CMS.EmailEngine.EmailMessage msg = new CMS.EmailEngine.EmailMessage();
    msg.From = "mymail@domain.com"; // use valid e-mail
    msg.Recipients = "mymail@domain.com"; // use valid e-mail
    msg.Subject = "Custom BizFrom e-mail";
    msg.Body = "The value of the FirstName field: "
                  + CMS.GlobalHelper.ValidationHelper.GetString(
                    this.viewBiz.BasicForm.DataRow["FirstName"], "N/A");
    CMS.EmailEngine.EmailSender.SendEmail(msg);
    lblConfirmationMessage.Text = "The e-mail has been sent.";
}
```

Please notice how you can retrieve the form values through the **viewBiz.BasicForm.DataRow** property.

Use valid e-mail addresses and **save** all changes.

5. Go to **CMS Desk -> Content**, choose the **Home** page, switch to the **Design** tab and add the **BizForm with custom e-mail** web part to the **zoneCenter** zone. Set the **Form name** property to the **Contact us** form (if you're using the sample Corporate Site).

6. Sign out and go to the live site. Enter some values into the form and submit it. You will see the additional confirmation message "The e-mail has been sent." and receive the e-mail.

You have seen how to create a modification of a standard web part.

7.18.3.5 Web part inheritance

Web part inheritance allows you to create a web part that has the same properties and uses the same code as the original web part, but has different default property values. It means you can create a specialized web part from a general one.

For example: You can create a news list web part inherited from the Repeater web part that will display a list of news by default. The default values can later be modified to any other value, but the inherited (specialized) web part allows you to do things faster.

How to create an inherited web part

1. Go to **Site Manager -> Development -> Web parts**, click **Listings and viewers** and click **New web part**.

2. Click **Inherit from an existing web part** and enter the following values:

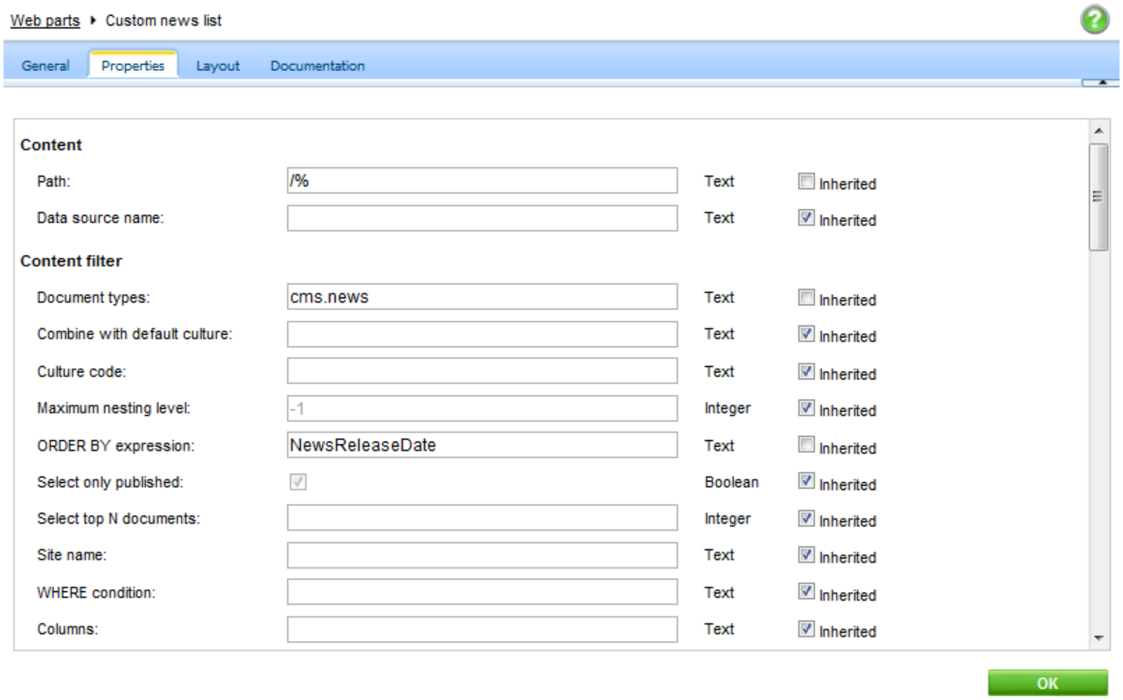
- **Display name:** Custom news list
- **Code name:** CustomNewsList
- **Inherit from:** Listings and viewers/Repeater

Click **OK**.

3. Switch to the **Properties** tab of the newly created web part. Here you can see the properties of the parent web part and you can override their default values by clearing the **Inherited** box and entering a new default value. Enter the following default values:

- **Path:** /%
- **Document types:** cms.news
- **ORDER BY expression:** NewsReleaseDate
- **Transformation:** cms.news.preview
- **Selected item transformation:** cms.news.default

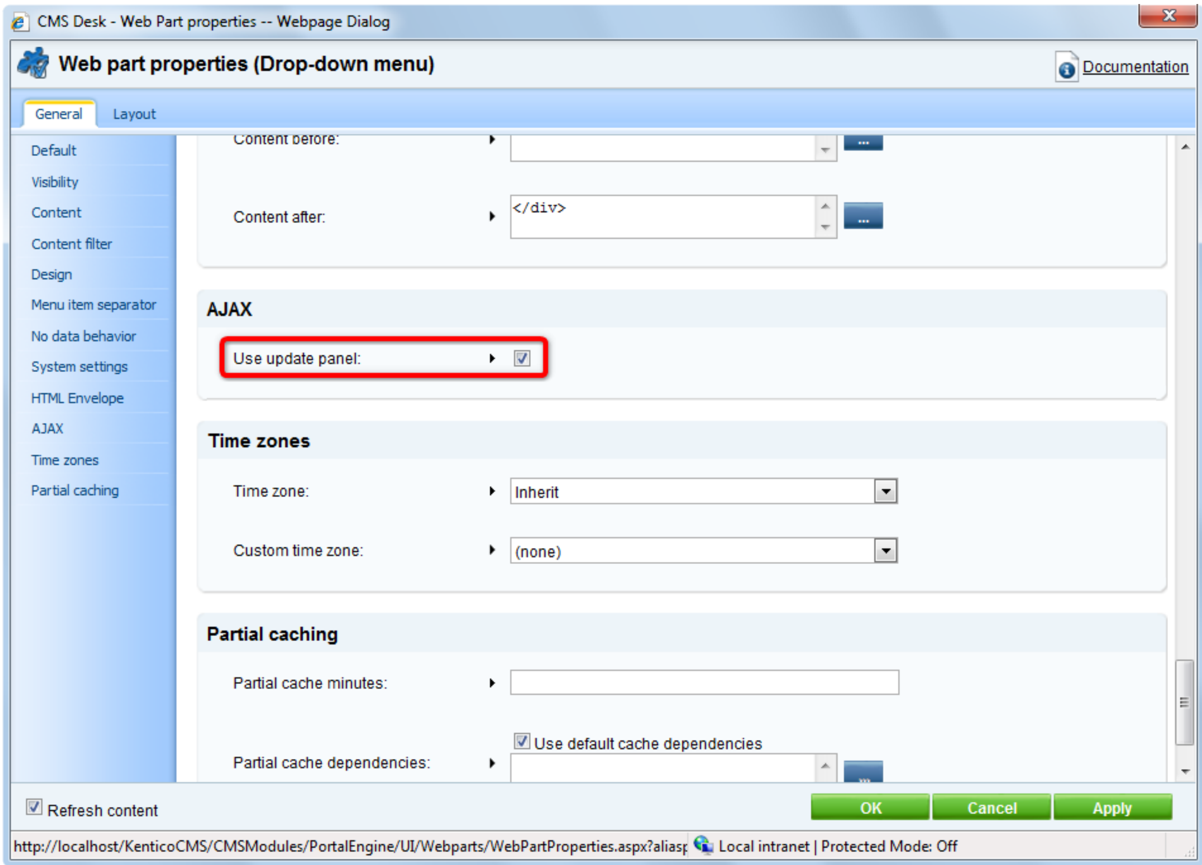
Click **OK**.



4. Go to **CMS Desk**, choose the **Home** page in the content tree and switch to the **Design** tab. Add the News list web part to the page. It will now display all site news without any additional configuration.

7.18.4 AJAX support

Your web parts can use the UpdatePanel, which wraps the web part into an AJAX UpdatePanel control. This can easily be done by enabling the **Use update panel** property of any web part.



No further modifications are necessary, but the following rules should be followed:

1. If the web part uses any dynamically loaded controls, their ID must be defined:

Incorrect:

```
Control ctrl = this.LoadControl("~/MyControl.ascx");
if (ctrl != null)
{
    Controls.Add(ctrl);
}
```

Correct:

```
Control ctrl = this.LoadControl("~/MyControl.ascx");
if (ctrl != null)
{
    ctrl.ID = "myControl";
    Controls.Add(ctrl);
}
```

2. When requesting PostBackEventReference, use Kentico's custom function instead of the default one:

Incorrect:

```
this.Page.ClientScript.GetPostBackEventReference(this, "");
```

Correct:

```
CMS.ExtendedControls.ControlsHelper.GetPostBackEventReference(this, "");
```

7.18.5 Web part containers

7.18.5.1 Containers overview

Containers are used as "boxes" for web parts. They consist of content that is displayed before and after the web part which means they are used as an envelope for web part content. They have three advantages over using the **ContentBefore** and **ContentAfter** properties:

1. They are re-usable for many web parts.
2. They can contain a title and dynamically inserted values of the web part properties.
3. They are objects in the system, which means they can be Exported and Imported.

Containers can be assigned not only to web parts, but also widgets and entire web part zones.

The containers can be managed at **Site Manager -> Development -> Web part containers**. The following properties can be defined when editing ( ) or creating a web part container:

Display name	The name of the container displayed to the users.
Code name	The name of the container used in code.
Text before web part	HTML content displayed before the web part - the beginning of the envelope.
Text after web part	HTML content displayed after the web part - the closing part of the envelope.

The **Text before/after web part** fields can contain dynamically inserted values of the web part properties. You can insert them using the following expression:

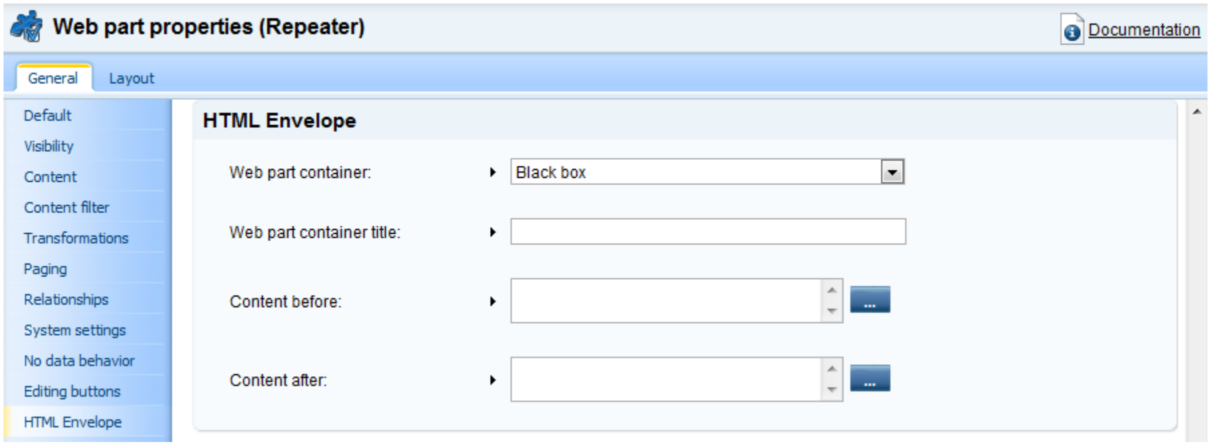
{%propertyname%}

You will most often use the following expression to insert the container title:

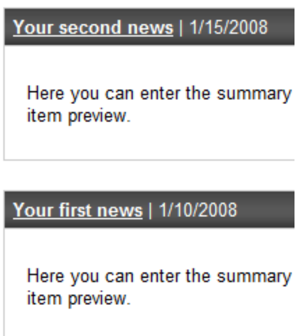
{%ContainerTitle%}

These macro expressions are resolved even when macro resolving for the particular web part (**Disable macros** property) is disabled.

Containers are assigned to web parts, widgets or zones simply by configuring ( ) them and selecting a container through the **Web part container** property.



Here's an example of a web part **without a container**:



Here's an example of the same web part **with a container**:

Please see the Creating web part containers topic for a step-by-step guide on how a new container is created.

7.18.5.2 Creating web part containers

Here's an example of defining a new web part container:

1. Go to **Site Manager -> Development -> Web part containers**.

2. Click **New container**.

3. Enter the following values:

- **Display name**: Blue box
- **Code name**: BlueBox
- **Text before web part**:

```
<table width="100%" style="border: 1px solid #4a62e4;" cellpadding="5" cellspacing
="0">
<tr valign="top">
<td style="background-color:#4a62e4;font-weight:bold;color:white">
{%ContainerTitle%}
</td>
</tr>
<tr valign="top">
<td>
```

- **Text after web part**:

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

4. Click **OK** to save.

5. Switch to the **Sites** tab and click the **Add sites** button to assign the new container to the website where you wish to use it.

6. Go to **CMS Desk** and display some page on the **Design** tab. **Configure** ( ) some web part and set its properties like this:

- **Use container**: Blue box
- **Container title**: My web part with container

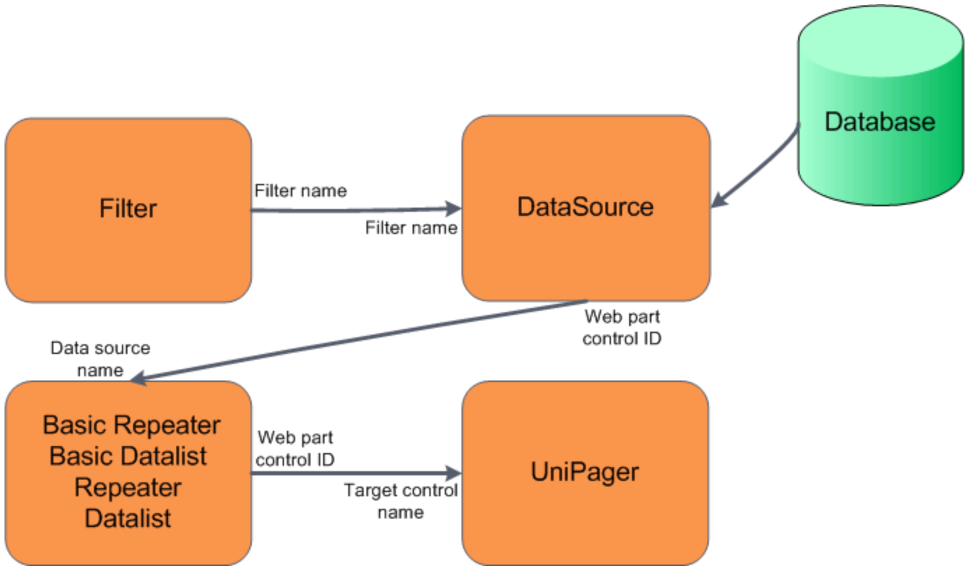
7. View the page in **Live site** mode. The web part will be surrounded by a blue border and it will have the specified title.

7.18.6 Data source web parts

7.18.6.1 Using Data source web parts

Data source web parts are designed for retrieving data from the database and sending it to other web parts that display it. This allows you to have separate web parts for retrieving data from the database, displaying data, filtering the displayed data and paging. These web parts can be placed anywhere on the page without any change in functionality and their design can be set separately, which results in higher design flexibility compared with using the original Repeater, Datalist, etc.

The following diagram shows how a Data source web part can be connected with other web parts to form a functional group of interconnected web parts. Captions of connecting lines show which properties of the web parts have to be set identically for the group to work properly.



Please note: Repeater and Datalist web parts can only be used for documents (provided by the **Documents data source** web part), not for e.g. custom tables etc. You will need to use the Basic repeater or Basic datalist for this purpose.

In the following example, we will create a group of interconnected web parts just as in the diagram above in order to see how the Data source concept works in practice.

1. Sign in to CMS Desk as the administrator (login *administrator* with blank password).

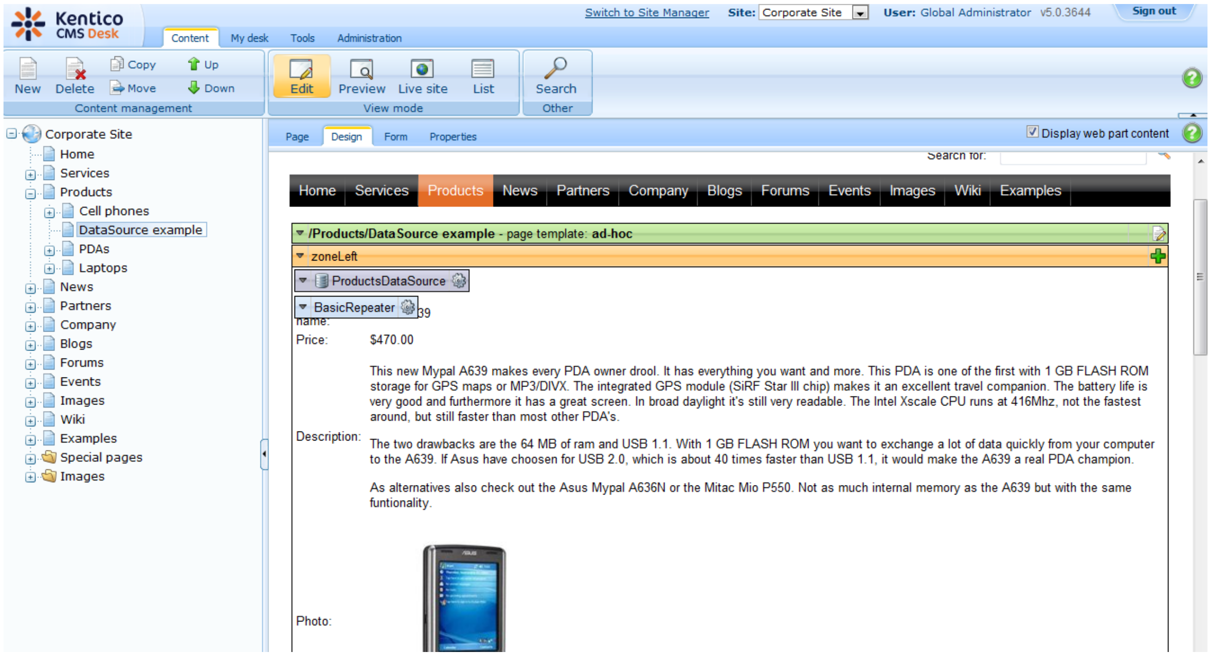
2. Create a new page under the **Products** section. Name it **DataSource example** and choose to **Create a blank page** using the **Simple** layout. Click **Save**.

3. Add the **Data Sources -> Products data source** web part to the only web part zone on the page. Leave default values for all its properties and click **OK**.

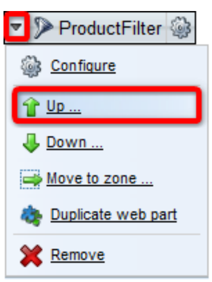
4. Add the **Listings & Viewers -> Basic Repeater** web part to the same web part zone. Set the following properties:

- **Data source name**: ProductsDataSource
- **Transformation name**: CMS.Product.Default

Click **OK**. You should see the repeater displaying products as in the following screenshot.



5. Now we will add the filter for the users to be able to filter displayed records. Add the **E-commerce -> Product filter** web part to the same web part zone. The web part will be added to the bottom of the page, so you may not be able to see it. You might want to scroll down and move the web part above the repeater. Open the web part's menu and choose to move the web part **Up**.



Set the following property of the filter web part:

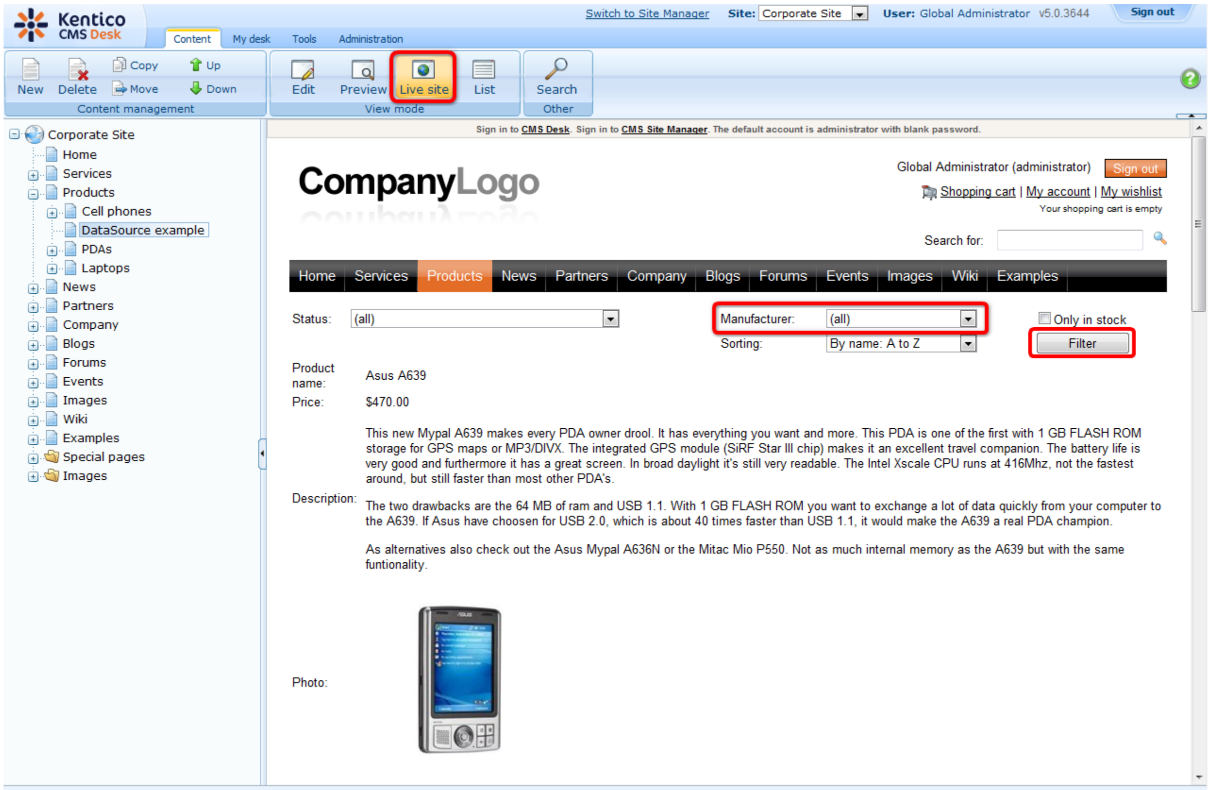
- **Filter name**: ProductFilter

Also open the properties of the Products data source web part and enter the same filter name into its **Filter name** property:

- **Filter name**: ProductFilter

You have just connected the data source with the filter. Data sent from the DataSource to the repeater can now be filtered using the ProductFilter.

6. To verify the functionality, switch to **Live site** mode. You can for example choose to display only products manufactured by Asus using the **Manufacturer** drop-down list. After selecting, click **Filter** for the changes to take effect. After doing so, you should see only Asus products in the repeater, as you can see in the screenshot below.



7. As the amount of data displayed by the repeater might grow very large in some cases, the next logical step is to add a pager. Add the **Listings & Viewers -> Universal pager** web part to the web part zone and move it **Up** above the repeater the same way that you moved the filter in step 5.

Set the following properties:

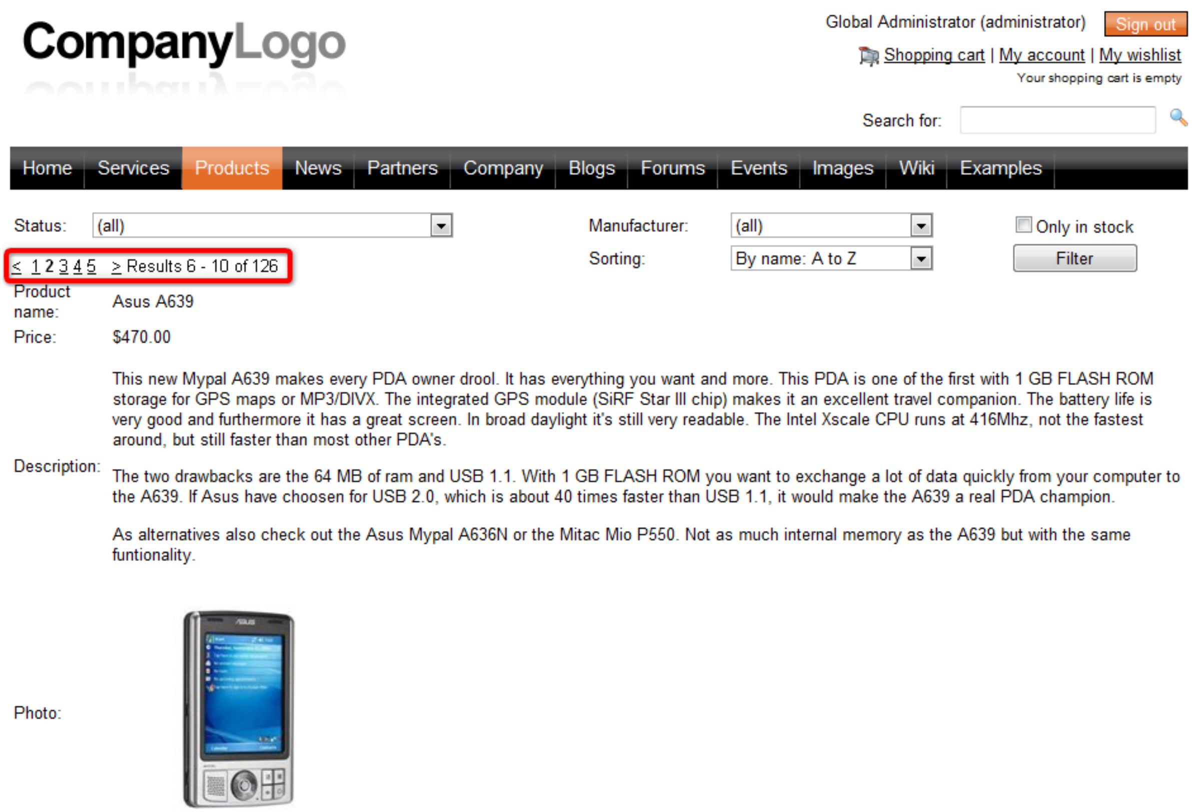
- **Target control name**: BasicRepeater
- **Page size**: 5
- **Group size**: 5

and make sure the following transformations are set:

- **Pages**: CMS.PagerTransformations.General-Pages
- **Current page**: CMS.PagerTransformations.General-CurrentPage
- **Previous page**: CMS.PagerTransformations.General-PreviousPage
- **Next page**: CMS.PagerTransformations.General-NextPage
- **Layout**: CMS.PagerTransformations.General-PagerLayout

Click **OK** to save the changes.

8. Now if you switch to **Live site** mode, the size of the page should be reduced to 5 products and you should be able to switch between pages using the pager.



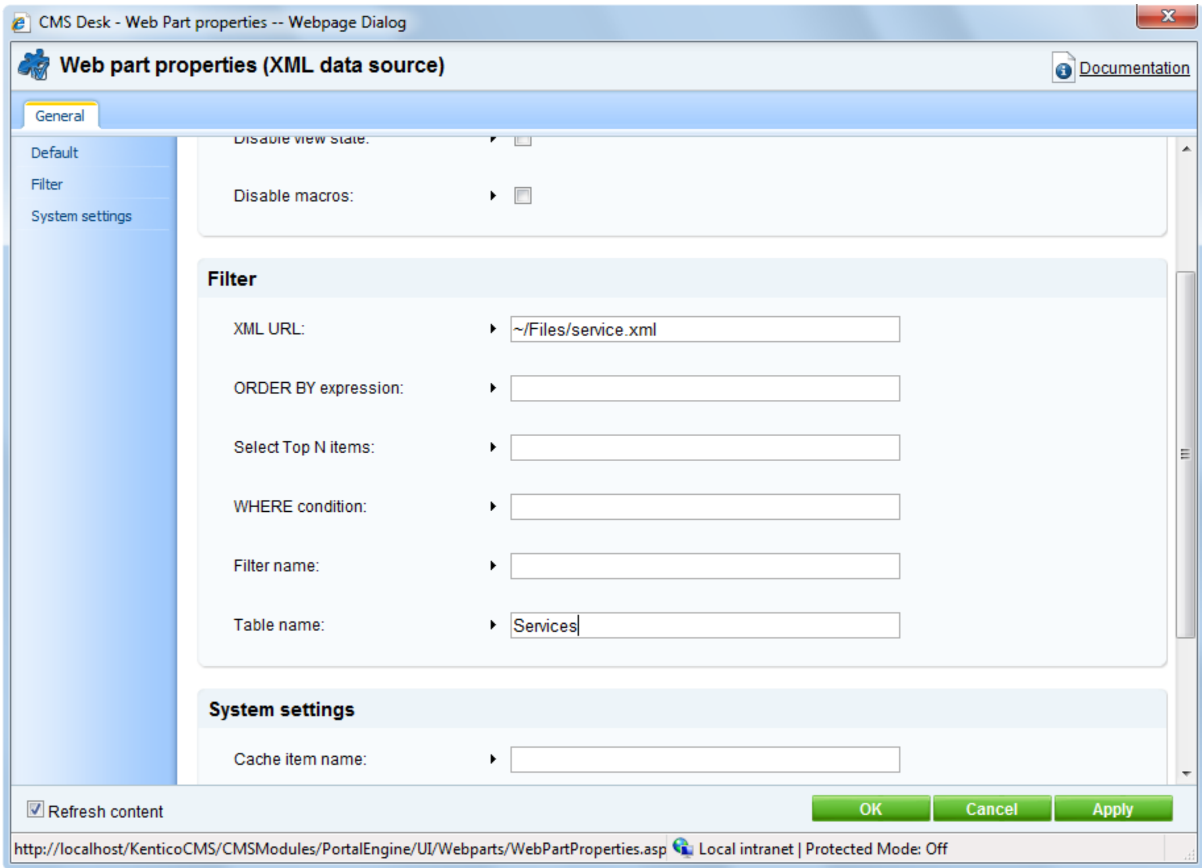
7.18.6.2 Problems with XML data source

The XML data source web part can be used to provide data from an XML file specified by its **XML URL** property. It uses the [Dataset.ReadXml()]() method to read the XML files. In some cases, this method can separate data from the source XML file into more than one dataset table. In this case, you need to specify which dataset table is used through the **Table name** property.

Unfortunately, it cannot be explicitly determined into which table the data that you require is loaded. Therefore, we recommend finding this out using the **Debug** function in **Visual Studio**.

If the **Table name** property is not specified appropriately in this case, no data is provided to the connected Repeater. It is therefore advisable to disable the Repeater's **Hide if no record found** property in order to prevent the repeater from being hidden on the page, which may be confusing in some cases.

In some special cases, it may also happen that the required data may be distributed into more than one dataset table. In such a case, the **Table name** property cannot be used to solve the issue and the only workaround is to modify the source XML (if possible).



7.18.6.3 Developing Data source web parts

The code example below shows the creation of a sample custom DataSource web part for providing users from the CMS. The web part consists of two controls, so you need to take the following two steps when developing your custom DataSource web part:

1. Create a user control that inherits from **CMSBaseDataSource**. This will be the control that gets the data.

- The **OnInit** and **DataFilter_OnFilterChanged** methods are necessary for handling the connected filter. If you are not planning to use a filter with the data source web part, you do not need to implement these methods.
- The **GetDataSourceFromDB** method ensures the loading of the actual data.

DataSourceControl.ascx

Please keep in mind that the following code is only an example, the actual values of the **CodeFile** and **Inherits** attributes will be different according to the name and location of the user control.

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="DataSourceControl.ascx.
cs" Inherits="CMSTestingSite_APIExamples_Controls_DataSourceControl" %>
```

DataSourceControl.ascx.cs

Please be aware that the name of the class will be different according to the name and location of the newly created user control.

[C#]

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using CMS.Controls;
using CMS.SiteProvider;

/// <summary>
/// Users data source. Loads all users with dependence on specified where
condition
/// </summary>
public partial class CMSTestingSite_APIExamples_Controls_DataSource :
CMSBaseDataSource
{
    #region "Variables"

    private string mWhereCondition = String.Empty;
    private string mOrderBy = String.Empty;

    #endregion


    #region "Properties"

    /// <summary>
    /// Gets or sets the where condition
    /// </summary>
    public string WhereCondition
    {
        get
        {
            return mWhereCondition;
        }
        set
        {
            mWhereCondition = value;
        }
    }


    /// <summary>
    /// Gets the data source
    /// </summary>
    public override object DataSource
    {
         get
         {
```

```
                return base.GetDataSource();
            }
        }

        /// <summary>
        /// Gets or sets the order by clause
        /// </summary>
        public string OrderBy
        {
            get
            {
                return mOrderBy;
            }
            set
            {
                mOrderBy = value;
            }
        }

        #endregion


        #region "Methods"

        /// <summary>
        /// Handle filter change event
        /// </summary>
        protected override void OnInit(EventArgs e)
        {
            if (SourceFilterControl != null)
            {
                SourceFilterControl.OnFilterChanged += new ActionEventHandler
                      (DataFilter_OnFilterChanged);
            }

            base.OnInit(e);
        }


        /// <summary>
        /// OnFilterChange handler
        /// </summary>
        void DataFilter_OnFilterChanged()
        {
            // Clear old data
            InvalidateLoadedData();
            // Raise change event
            this.RaiseOnFilterChanged();
        }


        /// <summary>
        /// Gets the datasource data
        /// </summary>
        public override object GetDataSourceFromDB()
        {
            // Check whether datasource contains data
```

```
        if (base.DataSource != null)
        {
            return base.DataSource;
        }
        // Initialize data and return datasource
        else
        {
            // Initialize properties with dependence on filter settings
            if (SourceFilterControl != null)
            {
                SourceFilterControl.InitDataProperties(this);
            }

            // Load data, in this example nothing is loaded
            base.DataSource = UserInfoProvider.GetFullUsers(this.WhereCondition,
this.OrderBy);
            return base.DataSource;
        }
    }

    #endregion
}
```

2. Create another user control and drop the first control on it. This control will become the actual web part.

- The control must inherit from **CMSAbstractWebPart**.
- The line of the **SetupControl** method beginning with *this.srcUsers.FilterName* ensures that the inner datasource's name is set the same as the *Web part control ID* property of this web part. Don't get perplexed by the property being called *FilterName*. This is by design, because datasources use the same objects as filters.

DataSourceWebPart.ascx

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="DataSourceWebPart.ascx.
cs" Inherits="CMSTestingSite_APIExamples_Controls_DataSourceWebPart" %>

<%@ Register src="DataSourceControl.ascx" tagname="DataSourceControl" tagprefix
="cms" %>

<cms:DataSourceControl ID="srcUsers" runat="server" />
```

DataSourceWebPart.ascx.cs

[C#]

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
using CMS.PortalControls;
using CMS.GlobalHelper;

public partial class CMSTestingSite_APIExamples_Controls_DataSourceWebPart :
CMSAbstractWebPart
{
    /// <summary>
    /// Gets or sets WHERE condition.
    /// </summary>
    public string WhereCondition
    {
        get
        {
            return ValidationHelper.GetString(this.GetValue("WhereCondition"), ""
);
        }
        set
        {
            this.SetValue("WhereCondition", value);
            srcUsers.WhereCondition = value;
        }
    }


    /// <summary>
    /// Gets or sets ORDER BY condition.
    /// </summary>
    public string OrderBy
    {
        get
        {
            return ValidationHelper.GetString(this.GetValue("OrderBy"), "");
        }
        set
        {
            this.SetValue("OrderBy", value);
            srcUsers.OrderBy = value;
        }
    }


    /// <summary>
    /// Gets or sets the source filter name
    /// </summary>
    public string FilterName
    {
        get
        {
            return ValidationHelper.GetString(this.GetValue("FilterName"), "");
        }
        set
        {
            this.SetValue("FilterName", value);
            srcUsers.SourceFilterName = value;
        }
    }
```

```
    /// <summary>
    /// Content loaded event handler
    /// </summary>
    public override void OnContentLoaded()
    {
        base.OnContentLoaded();
        SetupControl();
    }


    /// <summary>
    /// Initializes the control properties
    /// </summary>
    protected void SetupControl()
    {
        if (this.StopProcessing)
        {
            // Do nothing
        }
        else
        {
            this.srcUsers.WhereCondition = this.WhereCondition;
            this.srcUsers.OrderBy = this.OrderBy;

            // Sets the current web part name as datasource name
            this.srcUsers.FilterName = ValidationHelper.GetString(this.GetValue(
"WebPartControlID"), this.ClientID);
            // Sets the name of the extending filter
            this.srcUsers.SourceFilterName = this.FilterName;
        }
    }
}
```

You can now register this control as a web part in Kentico CMS as described in the Developing web parts topic.

7.18.6.4 Developing custom filters

Even though Kentico CMS comes with a built-in set of filter web parts for various different types of data, it may still in many cases be necessary to create a custom filter according to your specific requirements.

This can be achieved by implementing the filter as a *user control* that inherits from one of the following base classes:

- **CMSAbstractDataFilterControl** - works with document data sources.
- **CMSAbstractMenuFilterControl -** works with navigation web parts.
- **CMSAbstractQueryFilterControl -** works with custom table and query data sources.

These base classes can be found in the **CMS.Controls** namespace. The appropriate base class must be selected depending on the type of data to be filtered

When complete, the control can be placed onto your website through the **General -> Filter** web part. All that needs to be done is to specify the path to the *.ascx* file in the **Filter control path** property. The

Filter web part can then be attached to data source web parts of the matching type.

Of course, a custom filter can also be used anywhere in your code as a standard user control, for example on ASPX templates or in other web parts. All necessary properties, like the **FilterName** used to connect with the data source are inherited from the used base class.

Example

The example below demonstrates how a custom filter can be developed. This sample filter will work with *product documents*. It will support filtering according to product departments and allow the selection of several options that determine the order in which products are displayed. Filters for all types of documents or other objects can be created using the same approach.

1. Open the web project in Visual Studio and start by creating a **New folder** under the root called *CMSGlobalFiles* (if it doesn't already exist). The content of this folder can be exported along with your site when it is deployed to the live server.

2. Next, add a new **Web User Control** called *CustomProductFilter.ascx* into the **CMSGlobalFiles** folder and modify it to contain the following code:

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="CustomProductFilter.
ascx.cs" Inherits="CMSGlobalFiles_CustomProductFilter" %>

<table cellpadding="2">
  <tr>
    <td>
      <cms:LocalizedLabel ID="lblDepartment" runat="server" Text="Product
department" DisplayColon="true">
      </cms:LocalizedLabel>
    </td>
    <td>
      <cms:LocalizedDropDownList ID="drpDepartment" runat="server" Width="180" >
      </cms:LocalizedDropDownList>
    </td>
  </tr>
  <tr>
    <td>
      <cms:LocalizedLabel ID="lblOrder" runat="server" Text="Order by"
DisplayColon="true">
      </cms:LocalizedLabel>
    </td>
    <td>
      <cms:LocalizedDropDownList ID="drpOrder" runat="server" Width="180" >
      </cms:LocalizedDropDownList>
    </td>
  </tr>
  <tr>
    <td colspan="2">
      <cms:LocalizedButton ID="btnFilter" runat="server" Text="Apply Filter" />
    </td>
  </tr>
</table>
```

This creates the design of the filter's user interface. As you can see, it is composed of localized labels, drop-down lists and a button arranged in a simple table layout. Alternatively, it is possible to use a

CSS-based layout applied through HTML elements (e.g. <div>, <span>, etc.).

When developing a filter for actual live deployment, it may be preferable to enter the captions of the child controls using localization strings through the **ResourceString** property, rather than directly as **Text**.

3. Now switch to the code behind file of the user control, add the references shown below and set the control to inherit from the appropriate base class. This example uses the **CMSAbstractDataFilterControl** class, since the filter is intended for use with a document data source.

[C#]

```
using System;
using System.Data;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using CMS.Controls;
using CMS.GlobalHelper;
using CMS.Ecommerce;

public partial class CMSGlobalFiles_CustomProductFilter :
CMSAbstractDataFilterControl
{

    ...

}
```

The methods described in the following steps of this example will be added as members of this class.

4. Add the following three methods:

[C#]

```
/// <summary>
/// Setup the inner child controls
/// </summary>
private void SetupControl()
{
    // Hide the filter if StopProcessing is enabled
    if (this.StopProcessing)
    {
        this.Visible = false;
    }

    // Initialize only if the current request is NOT a postback
    else if (!RequestHelper.IsPostBack())
    {
        // Loads product departments as filtering options
        InitializeDepartments();

        // Initialize the order by drop-down list
```

```
        IntializeOrder();
    }
}


/// <summary>
/// Ensures that the ordering options are loaded into the order by drop-down list
/// </summary>
private void IntializeOrder()
{
    // Initialize options
    this.drpOrder.Items.Add(new ListItem("Price - Ascending", "priceAsc"));
    this.drpOrder.Items.Add(new ListItem("Price - Descending", "priceDesc"));
    this.drpOrder.Items.Add(new ListItem("Product name", "name"));
}


/// <summary>
/// Loads all existing product departments as filtering options into the
department drop-down list
/// </summary>
private void InitializeDepartments()
{
    // Get all product departments
    DataSet departments = DepartmentInfoProvider.GetDepartments("",
"DepartmentDisplayName", 0, "DepartmentID, DepartmentDisplayName");

    // Check if there are any product departments
    if (!DataHelper.DataSourceIsEmpty(departments))
    {
        // Bind deparments to the drop-down list
        this.drpDepartment.DataSource = departments;
        this.drpDepartment.DataTextField = "DepartmentDisplayName";
        this.drpDepartment.DataValueField = "DepartmentID";

        this.drpDepartment.DataBind();

        // Add default '(all)' value
        this.drpDepartment.Items.Insert(0, new ListItem("(all)", "##ALL##"));
    }
}
```

These ensure that the correct filtering options are loaded into the child drop-down lists.

5. The most important part of the filter is performed by the **SetFilter()** method. Define it as shown below:

[C#]

```
/// <summary>
/// Generates a WHERE condition and ORDER BY clause based on the current filtering
selection
/// </summary>
private void SetFilter()
{
    string where = null;
```

```
    string order = null;

    // Generate a WHERE condition based on the selected product department
    if (this.drpDepartment.SelectedValue != null)
    {
        int departmentId = ValidationHelper.GetInteger(this.drpDepartment.
SelectedValue, 0);

        if (departmentId > 0)
        {
            where = "SKUDepartmentID = " + departmentId;
        }
    }

    // Apply the selected ordering direction
    if (this.drpOrder.SelectedValue != "")
    {
        switch (this.drpOrder.SelectedValue)
        {
            case "priceAsc":
                order = "SKUPrice";
                break;

            case "priceDesc":
                order = "SKUPrice Desc";
                break;

            case "name":
                order = "SKUName";
                break;
        }
    }

    if (where != null)
    {
        // Set where condition
        this.WhereCondition = where;
    }

    if (order != null)
    {
        // Set orderBy clause
        this.OrderBy = order;
    }

    // Raise the filter changed event
    this.RaiseOnFilterChanged();
}
```

The custom filter control dynamically generates a WHERE condition and ORDER BY statement based on the current filter selection, which is then used to set the value of the **WhereCondition** and **OrderBy** members inherited from the base class. These members are read by the data source to which the filter is attached and inserted into the SQL query used to load data. Remember that the filter must inherit from the appropriate base class according to the type of the used data source.

6. Finally, add two methods that override the handlers of the **Init** and **PreRender** events as shown below:

[C#]

```
/// <summary>
/// Init event handler
/// </summary>
protected override void OnInit(EventArgs e)
{
    // Create child controls
    SetupControl();

    base.OnInit(e);
}


/// <summary>
/// PreRender event handler
/// </summary>
protected override void OnPreRender(EventArgs e)
{
    // Check if the current request is a postback
    if (RequestHelper.IsPostBack())
    {
        // Apply the filter to the displayed data
        SetFilter();
    }

    base.OnPreRender(e);
}
```

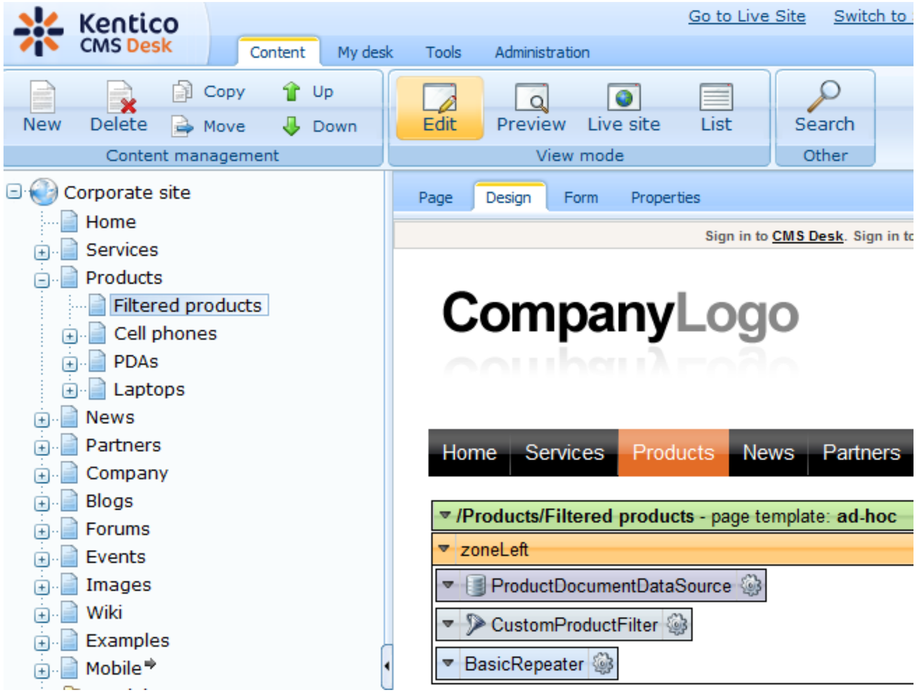
The handlers ensure that the appropriate private methods (defined previously in the example) are called during the correct stages of the page life cycle.

When the *Apply Filter* button is clicked, a postback occurs, which triggers the **SetFilter()** method during the **PreRender** event and the filter is applied to the displayed data. Notice the use of the **CMS.GlobalHelper.RequestHelper.IsPostBack()** method in the conditions. The **SetFilter()** method is only called when the current page request is a postback and the child controls are initialized only when this is not the case.

Save the changes to both files.

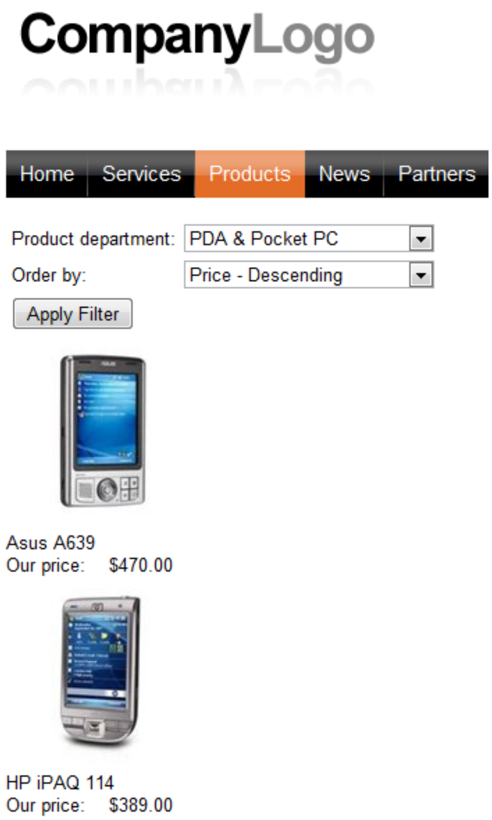
7. Now that the filter control is finished, you can try out its functionality. It is recommended to test the filter on the **Corporate** or **E-commerce** sample sites, since they already contain examples of product documents. Go to **CMS Desk**, create a new page and place the following web parts on it:

- **Documents data source** - configure it to load product documents and fill in its **Filter name** property (e.g. *CustomProductFilter).*
- **Filter** - configure a matching **Filter name** (*CustomProductFilter*) and enter the path of the .ascx file implementing your custom filter into the **Filter control path** property (*~/CMSGlobalFiles/CustomProductFilter.ascx*).
- **Basic repeater** - enter the ID of the Documents data source web part into the **Data source name** property and set an appropriate **Transformation name**, for example *Ecommerce.Transformations.Product_SimplePreview*.



Detailed information about how this kind of web part combination works can be found in the Using Data source web parts topic.

8. If you open the page on the live site, you will see a list of products with the custom filter displayed above. The products can be filtered by their department and the order in which they are displayed can be changed.



7.19 Widgets

7.19.1 Overview

Widgets introduce support for the personalization of pages. This feature allows users and editors to edit the structure of page templates. The personalized settings are saved within the system and can be invoked from the live site by authorized users or through the CMS Desk interface in the case of website editors.

From a designer's point of view, widgets are the basic building blocks of page templates in the same way as web parts. Like web parts, they are placed into zones, which must be configured to contain a certain type of widgets. For more information about web development basics, such as page templates, please refer to the Web development overview -> Portal engine development model chapter of this guide.

There is a predefined set of widgets in the default CMS Kentico installation. As all widgets are based on existing web parts, you can alternatively create your own widgets. Widgets are created and managed at **Site Manager** -> **Development** -> **Widgets**. The details are given in the Developing widgets topic.

As mentioned above, widgets provide the same functionality as ordinary web parts. However, users with appropriate rights can modify widget properties, their placement on the web page, add and remove widgets from their pages and so on. Users are divided into four different groups according to the possible ways they can use widgets:

- Site developers/Administrators - define the placement of widget zones and their default content, select the web part properties available for personalization in widgets and manage all available widgets

- Page editors - define the content of widget zones created on the page templates by the site developers/administrators
- Group administrators - define the content of widget zones on group page templates
- Website users - customize the design of their home page or another page which can be personalized on the live site

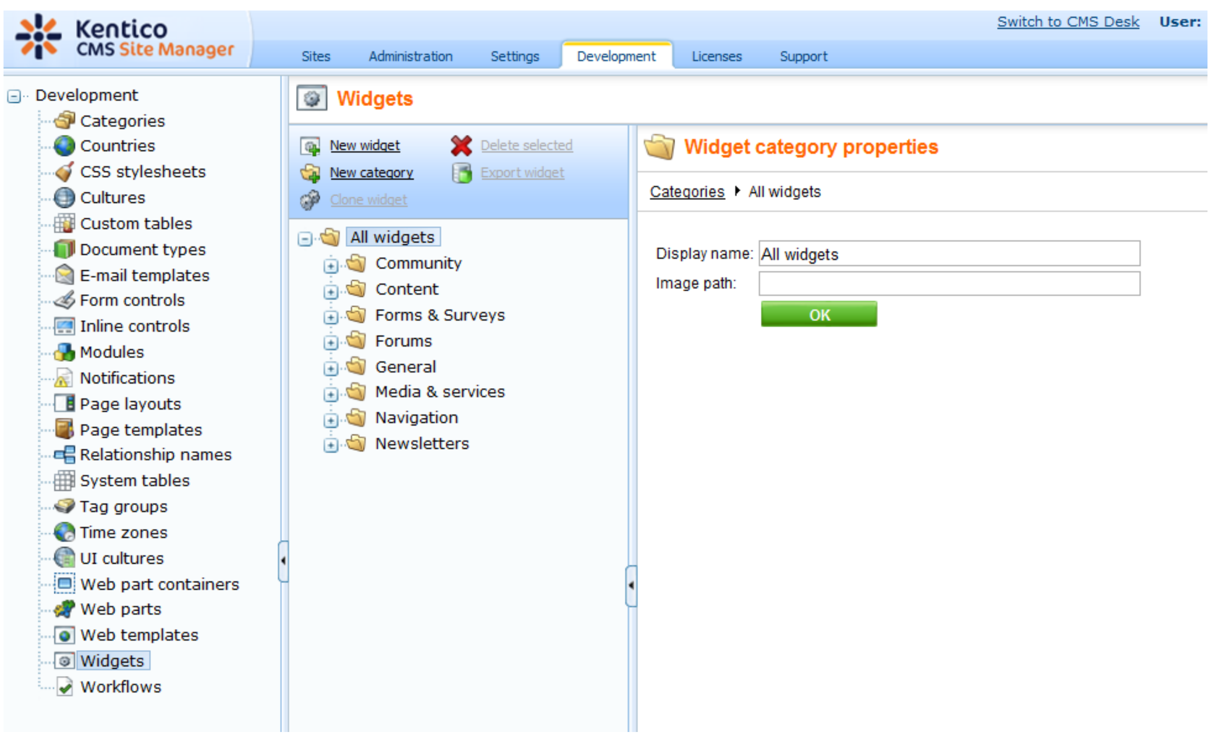
Security issues, such as the definition of where individual widgets can be used and permissions for different types of roles are discussed in the Security topic.

The Widget internals and API sub-chapter provides information about the database tables and classes used by widgets and examples of how widgets can be managed using the API.

7.19.2 Developing widgets

Every widget is based on an existing web part. As a site developer, you can simply create your own widgets. This topic will lead you through the step-by-step process of creating a widget.

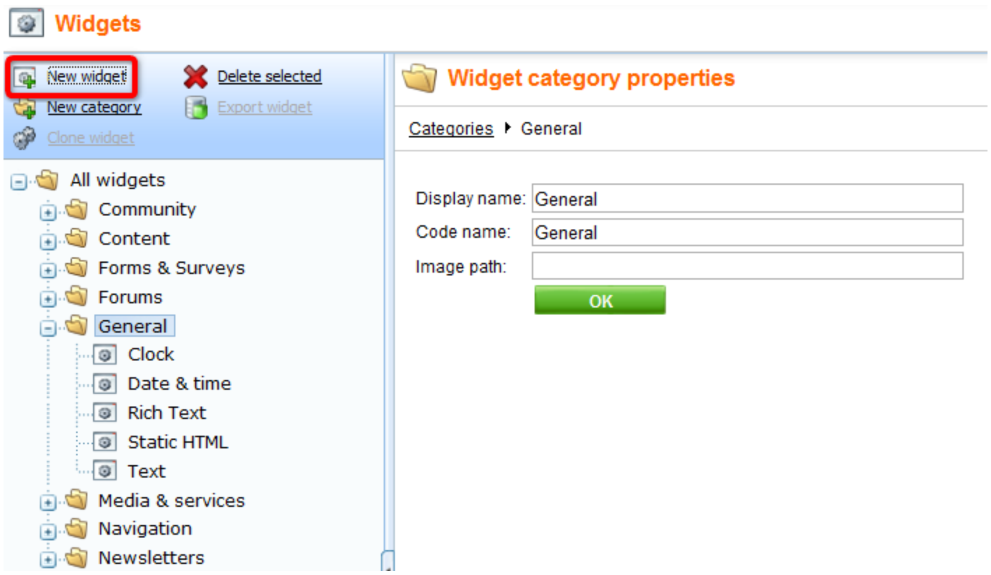
1. As mentioned above, it is not possible to create a widget without a reference to a web part. To start, switch to **Site Manager** -> **Development** -> **Widgets**, for the administration interface where widgets can be created. You should see a screen similar to the following one.



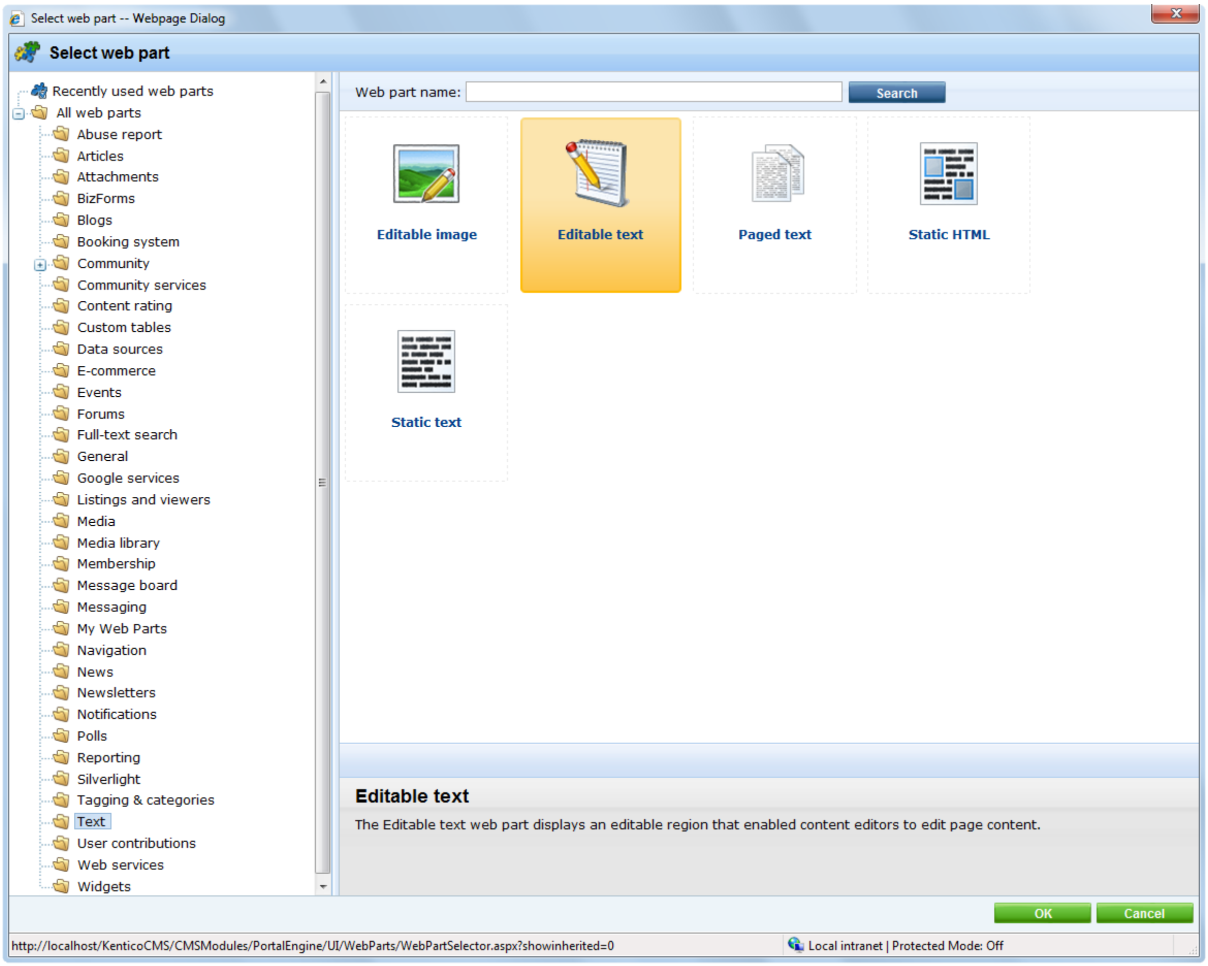
2. Widgets are grouped into categories. It is important to note that categories are used for the sake of organization only. The category under which a given widget is stored has no influence on who can use or modify the widget.

Clicking on a node causes the selected category to open and all widgets stored within that category are listed. You can also change the **Display** and **Code name** of a category. The **Image path** property sets the path to the image which can be used as the category icon in the **Add widget** dialog (the recommended size is 16x16px).

Now select any category where the new widget will be stored and click the **New widget** link. Alternatively, you can create the new widget under any folder and change its location later when the properties of the widget are defined.



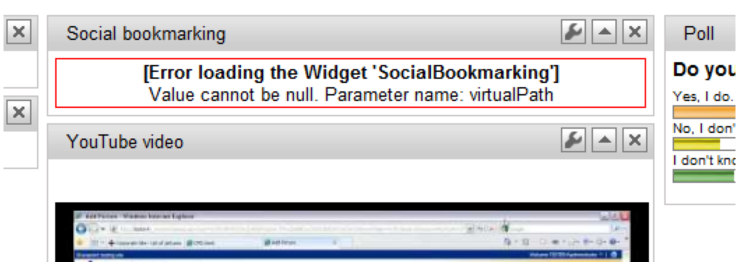
3. A dialog opens, where you can select a web part to be used as a template for the new widget. In the example below the **Editable text** web part from the **Text** category is selected. To confirm the selection click the **OK** button.



4. A new widget has just been created. You can manage widgets using the actions above the tree. A new widget can be created based on an existing one using the **Clone widget** link. If you are about to create a new category use the **New category** link.

To delete a selected widget or the whole category (except the *All widgets* category), click the **Delete selected** link. Please bear in mind that when a widget is deleted, it is not automatically removed from page templates. Therefore, when pages containing a deleted widget are to be displayed, an error message will be displayed instead of the missing widget.

The example below shows the effect of deleting the **Social bookmarking** widget without removing it from the page templates.

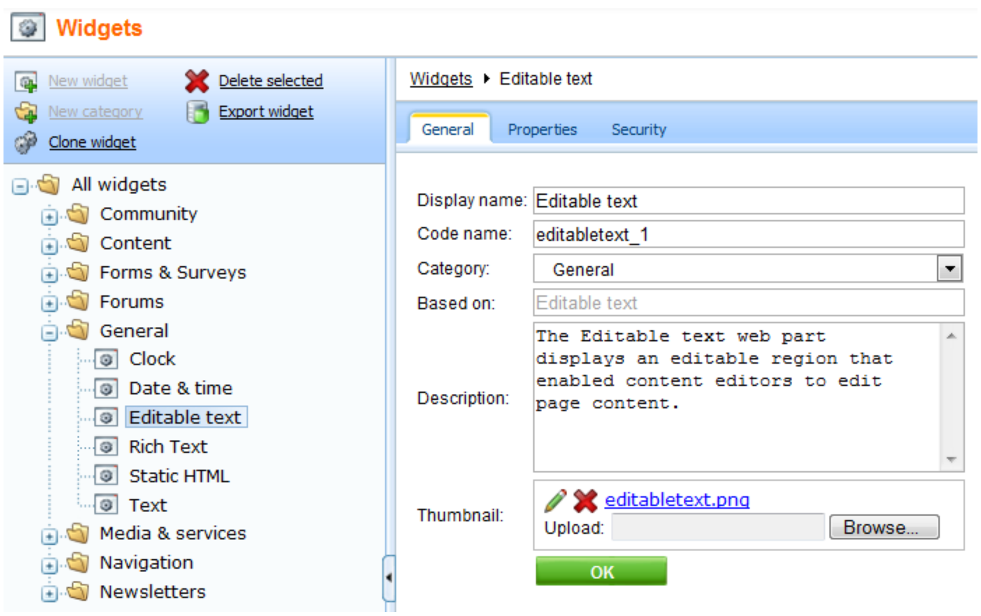


A selected widget can also be exported by clicking on the **Export widget** link. For details regarding export see Exporting single objects.

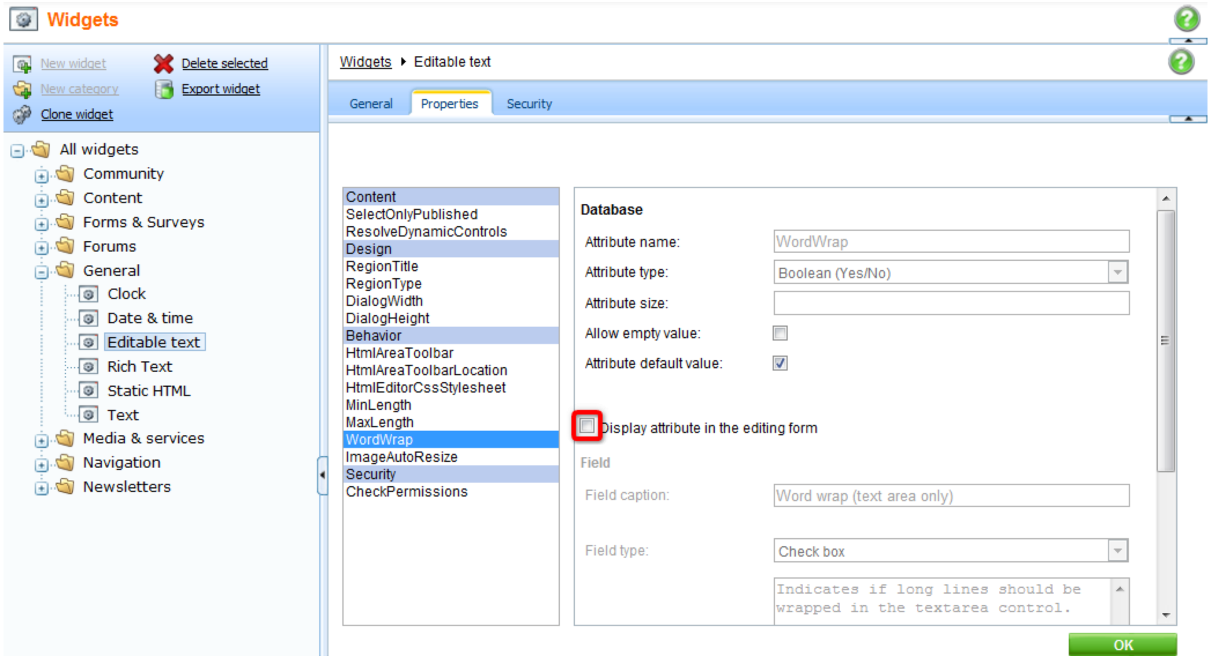
5. For each selected widget there are three tabs available. On the **General** tab there are all the basic settings:

- **Display name** - descriptive name of the widget used in the Kentico CMS user interface
- **Code name** - name of the widget used in website code
- **Based on** - the name of the web part that the widget is derived from
- **Category** - defines to which category the widget belongs; you can change it here by selecting a new category from the drop-down list
- **Description** - text description of the widget
- **Thumbnail** - graphical representation of the widget in the widget selection dialog

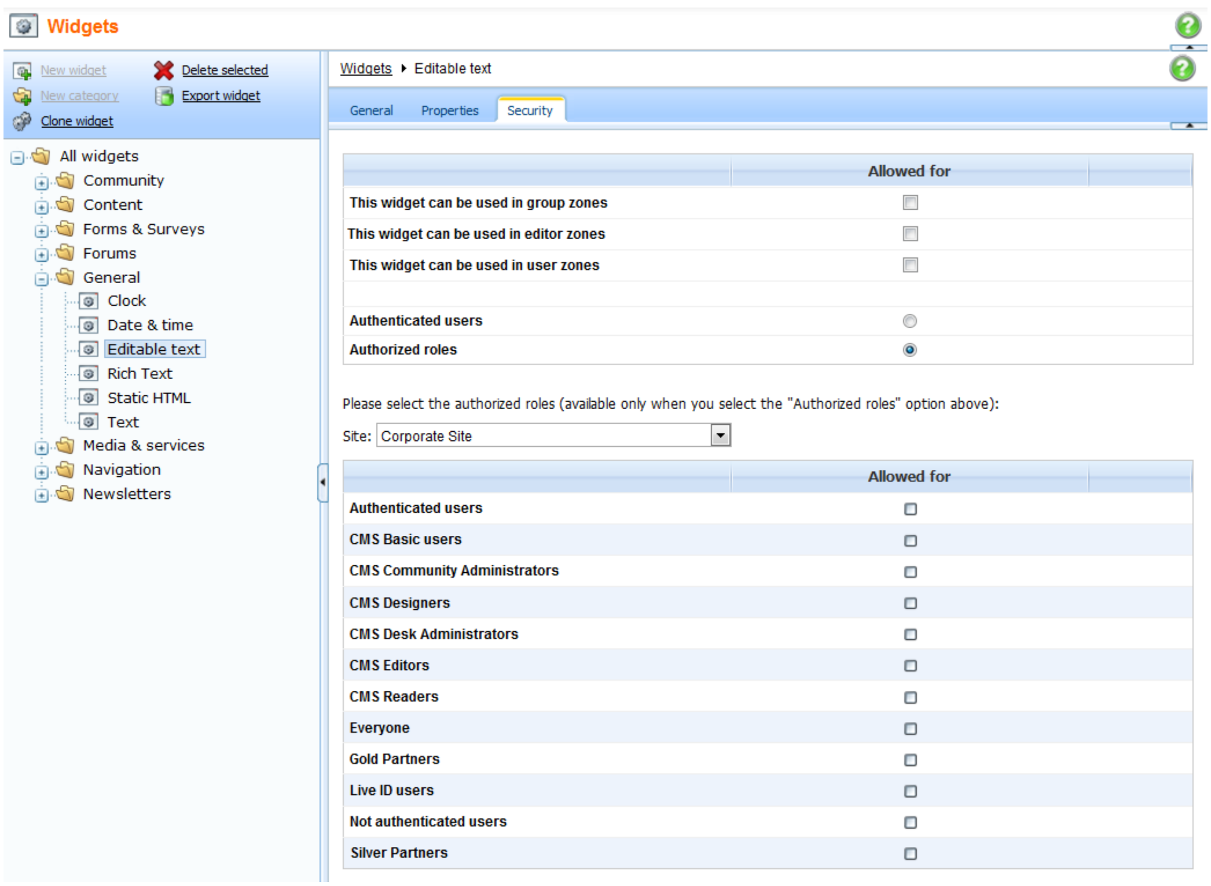
Do any changes required and click the **OK** button.



6. The **Properties** tab includes all the properties available for the parent web part. You can notice that all of them are disabled by default. Using the **Display attribute in the editing form** check box you can choose which property should be available to the widget's users. Please bear in mind that any changes must be confirmed by clicking the **OK** button.



7. On the **Security** tab there are settings specifying where the widget can be used and who will be able to modify it. The details are given in the Widgets security topic.



8. When all settings are done the widget is ready for use in a page template.

7.19.3 Using widgets

There are different sets of widget related actions that users can perform depending on their role as shown below:

- Site developers/Administrators - Can determine which zones are widget zones (and their type) and which ones are standard (static) web part zones, and also define the default content (layout) of these zones. These tasks can be done at **CMS Desk -> Content -> Design**. Can also manage widgets in the administration interface at **Site Manager -> Development -> Widgets**. This includes developing new widgets, determining where and by whom they can be used and selecting the properties which should be personalizable (editable by users).
- Page editors - Can define the content of Editor widget zones created on the page templates by the site developers/administrators. This is accessible through **CMS Desk -> Content -> Edit**.
- Group administrators - Can define the design of Group administrator widget zones on group page templates. Group administrators can add or remove widgets and set their properties just like page editors. This customization can be done on the live site.
- Website users - Authenticated users of the website can customize the design of their home page or another page which can be personalized on the live site. They can add or remove widgets and set the widget properties. Only pages with templates that contain User widget zones can be personalized in this way.

Site developers/Administrators

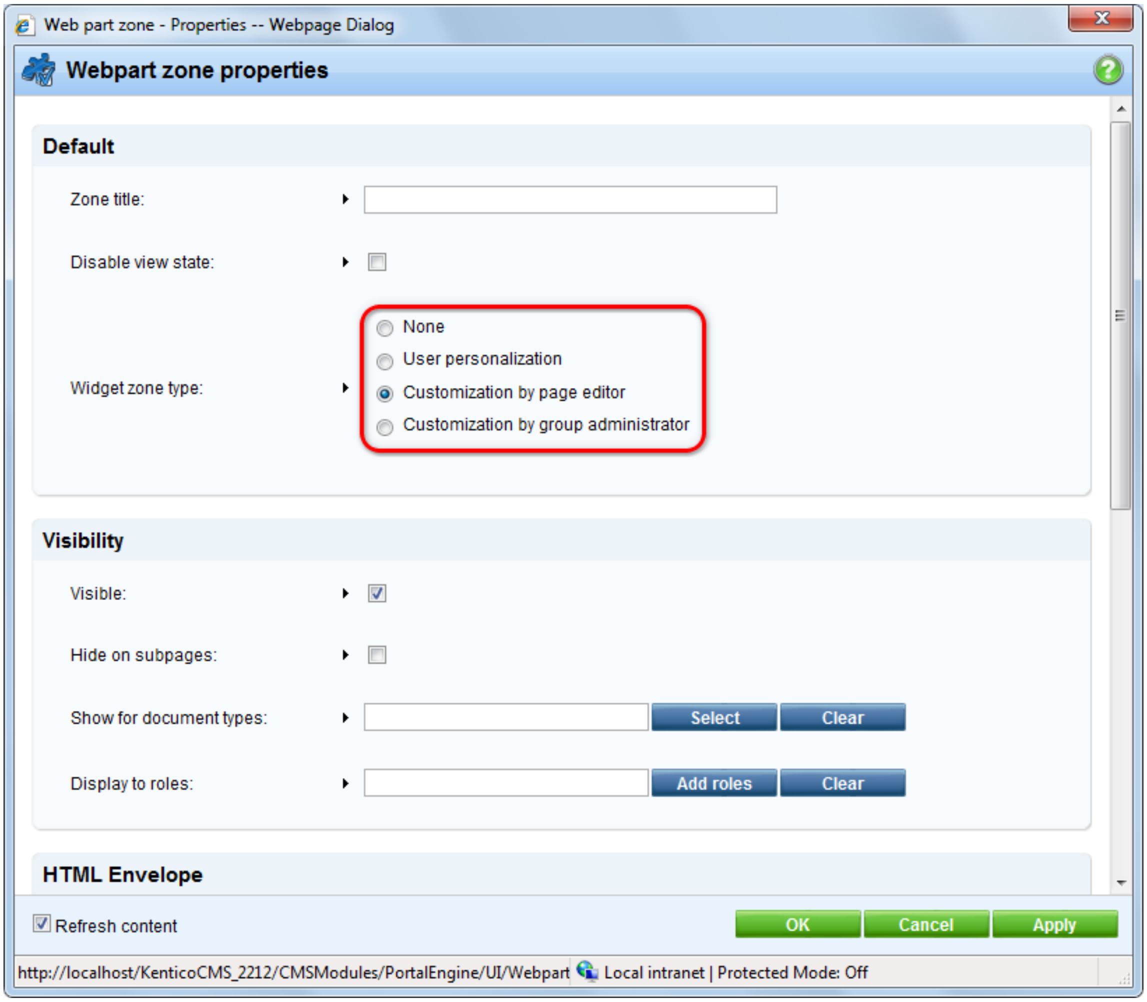
The site developers make the decisions regarding the layout of page templates. This also includes the decision whether a zone will be defined as a web part zone (static) or widget zone (customizable).

To change a web part zone into a widget zone, go to **CMS Desk -> Content -> Design**, expand the menu of the selected web part zone (▾) and click on the **Properties** item.

Using the **Web zone part properties** dialog you can define the **Widget zone type** property. The following options are available:

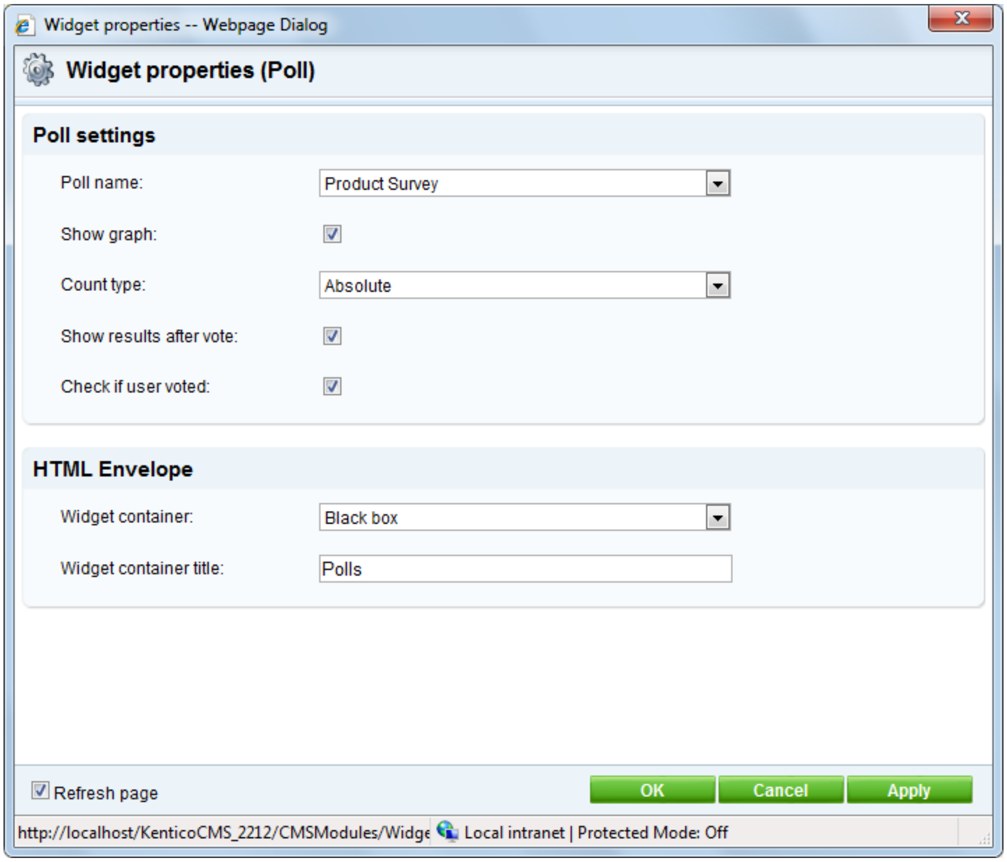
- **None** - a standard web part zone is used
- **User personalization** - the zone will be a widget zone and website users will be able to personalize it on the live site; the zone will be marked with the **User zone** ( ) icon
- **Customization by page editor** - the zone will be a widget zone and website editors will be able to customize it; the zone will be marked with the **Editor zone** ( ) icon
- **Customization by group administrator** - the zone will be a widget zone and group administrators of the group that owns the page will be able to customize it; the zone will be marked with the **Group administrator zone** ( ) icon

Please bear in mind that changing the widget zone type removes all current web parts or default widgets placed in that zone.



Having defined the zone as a widget type zone, you can now add widgets. Simply click on the **Add widget** () button on the right side of the zone. You can select any widget available for the given type of widget zone (User, Editor, Group administrator).

A **Widget properties** dialog opens, where you can change any properties available for customization. To save any changes, just click on the **OK** button.



This way, developers can define the default content of widget zones. It is up to the users to personalize these zones further as is described in the remaining sections of this topic.

Page editors

As a page editor, you can add, modify or delete any widget placed in editor type widget zones. This can be used to allow editors to quickly customize pages beyond simply entering content into static page templates.

To add a widget, just click on the **Add widget** ( ) button in the top left corner of an editor widget zone.

You can modify any widget placed in an editor zone by clicking on the **Configure widget** ( ) button. You can also delete widgets using the **Delete widget** ( ) button. Widgets can be moved between different widget zones by clicking on the **Drag widget** ( ) button and dragging the widget to the desired place.

Group administrators

Group administrators (members of the group who are assigned to a group role that can manage the group) have basically the same options as page editors. They can personalize those pages that belong to their group that contain group administrator widget zones by adding, deleting and configuring widgets in these zones. This is done on the live site, as group administrators won't necessarily have access to the **CMS Desk -> Edit** interface. Other website users who have access to the group pages will be able to see the widgets, but won't be able to manage them.

For specific information about groups, please see the Modules -> Groups chapter of this guide.

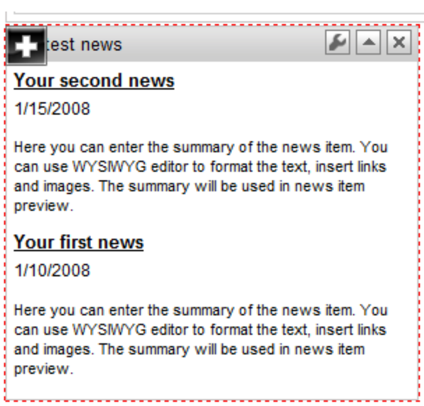
Please note

Group ownership of a document can be configured at **CMS Desk -> Edit -> Properties -> General -> Owned by group**.

Website users

Authenticated website users can manipulate user widget zones on the live site and thus personalize their pages. A common use is placing user widget zones onto the home page, so that users can create their own personalized version of it.

In order to add a widget to a user widget zone on the live site, click on the **Add widget** ( ) button in the top left corner of the zone.



Any widget can be configured by clicking on the **Configure widget** ( ) button. You can also delete widgets using the **Delete widget** ( ) button. To move the widgets between zones, click on the **Drag widget** ( ) button and drag the widget to the new zone. Any widget can be minimized using the **Minimize** ( ) button, or restored using the **Maximize** ( ) button.

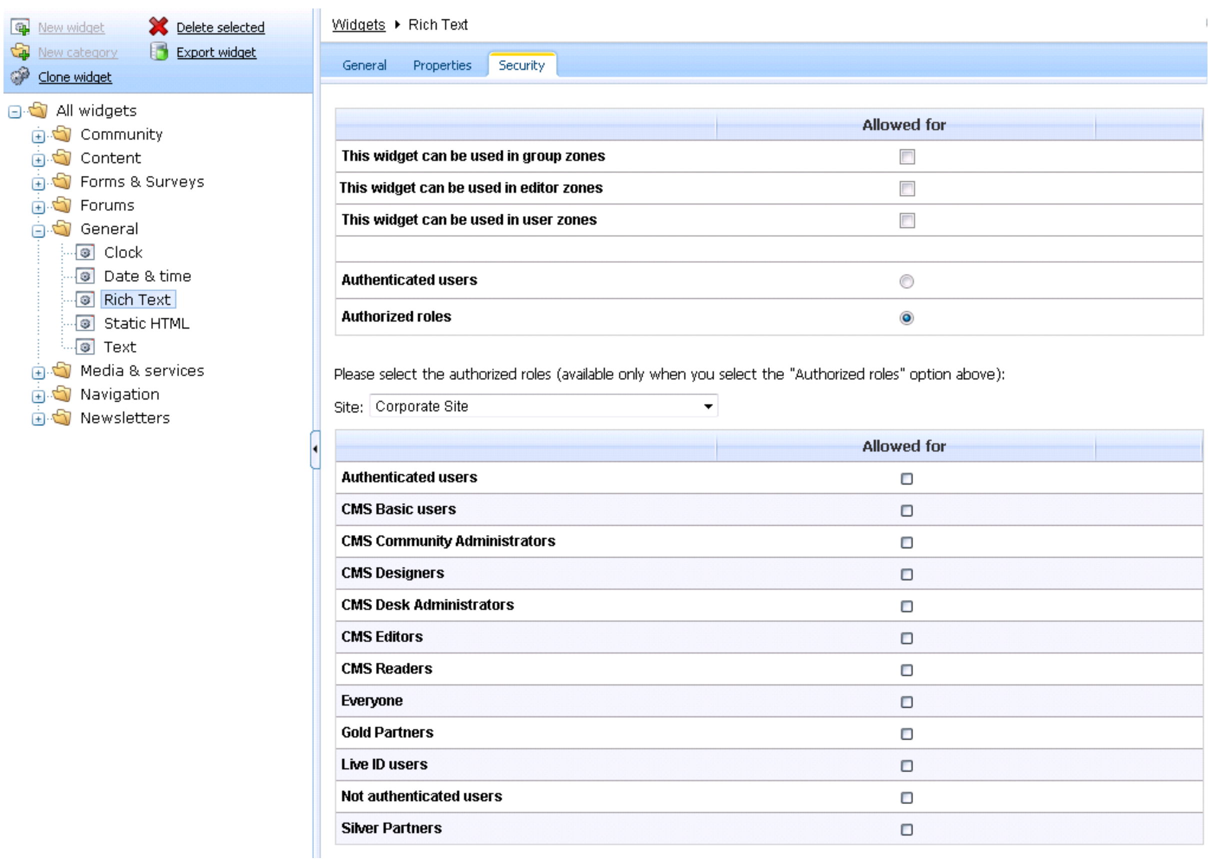
Widget actions web part

If there is a **Widget actions web part** on the page, users can add widgets or reset all widget zones on the page to their default setting by clicking on the appropriate links. The web part can only work with the type of widget zones specified by its **Widget zone type** property. When adding a widget using the **Add widget** link, the widget is placed into the zone specified by its **Widget zone** property. Users can drag the widget into another zone if the default placement is not suitable.

7.19.4 Security

When a new widget is created from an existing web part, all the attributes are hidden in the editing form by default. It is up to the widget creator to select attributes which should be available for customization using check boxes on the **Properties** tab.

The security options are defined on the **Security** tab when selecting a widget from the content tree at **Site Manager -> Development -> Widgets**. By default, all widgets are forbidden for all zone types and are allowed for authorized roles only. However, no authorized role is selected by default. It is up to the developers to allow the widget for a specific type of zone and role.



Please keep in mind that changing the security settings will affect new widgets only. If a user was allowed to add a widget and an administrator later removes this permission, the user can still see the widget on their page. However, once deleted, the widget cannot be added back to the page without allowing it on the **Security** tab of that particular widget.

7.19.5 Widget internals and API

7.19.5.1 Database tables and API classes

The following database tables are used to store widget data:

- **CMS_Widget -** contains widgets and their settings
- **CMS_WidgetCategory -** contains widget categories and information about them
- **CMS_WidgetRole -** assigns roles and their permissions to widgets

The data about the configuration and placement of individual instances of widgets on page templates is stored in the *CMS_PageTemplate* table.

The widgets API is provided by the following **CMS.PortalEngine** namespace classes:

- **WidgetInfo**, **WidgetInfoProvider -** these classes provide functionality for managing widgets
- **WidgetCategoryInfo**, **WidgetCategoryInfoProvider -** this class provides functionality for managing widget categories

The following topics show examples of how these classes can be used:

- Getting widget data
- Managing widgets and their categories
- Setting widget permissions

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

7.19.5.2 Getting widget data

The following sample code shows how you can get a widget as a *WidgetInfo* object:

[C#]

```
using CMS.PortalEngine;

...

        // Get widget object by ID
        WidgetInfo wi = WidgetInfoProvider.GetWidgetInfo(1);

        // Get widget object by code name
        WidgetInfo wi2 = WidgetInfoProvider.GetWidgetInfo("TestWidget");
```

The following sample code shows how you can get a DataSet containing widgets from the system:

[C#]

```
using System.Data;
using CMS.PortalEngine;

...

        string where = "WidgetCategoryID = 76";
        string orderBy = "WidgetDisplayName";
        int topN = 10;
        string columns = "WidgetID, WidgetDisplayName, WidgetCategoryID";

        // Get dataset of widgets according to the given parameters
        DataSet ds = WidgetInfoProvider.GetWidgets(where, orderBy, topN, columns);
```

7.19.5.3 Managing widgets and their categories

The following sample code shows how you can create a new widget category under the root category and add it to the system:

[C#]

```
using CMS.PortalEngine;

...

        // Get parent category
        WidgetCategoryInfo parent = WidgetCategoryInfoProvider.GetWidgetCategoryInfo

        // If parent exists
        if (parent != null)
        {
            // Create new category object
            WidgetCategoryInfo wci = new WidgetCategoryInfo();

            // Set properties
            wci.WidgetCategoryDisplayName = "Example category";
            wci.WidgetCategoryName = "ExampleCategory";
            wci.WidgetCategoryParentID = parent.WidgetCategoryID;

            // Save to database
            WidgetCategoryInfoProvider.SetWidgetCategoryInfo(wci);
        }
```

The following sample code shows how you can create a new widget based on the **Abuse report** web part under an existing category and add it to the system:

[C#]

```
using CMS.PortalEngine;
using CMS.FormEngine;
```

```
...

        // Get parent web part and category for widget
        WebPartInfo wpi = WebPartInfoProvider.GetWebPartInfo("AbuseReport");
        WidgetCategoryInfo category = WidgetCategoryInfoProvider.GetWidgetCategoryIn

        // If parent web part and category exist +
        // Parent web part must not be inherited - widgets cannot be created from an
        if ((wpi != null) && (wpi.WebPartParentID == 0) && (category != null))
        {
            // Create new widget object
            WidgetInfo wi = new WidgetInfo();

            // Set properties from parent web part
            wi.WidgetName = wpi.WebPartName;
            wi.WidgetDisplayName = wpi.WebPartDisplayName;
            wi.WidgetDescription = wpi.WebPartDescription;

            wi.WidgetProperties = FormHelper.GetFormFieldsWithDefaultValue(wpi.
WebPartProperties, "visible", "false");

            wi.WidgetWebPartID = wpi.WebPartID;
            wi.WidgetCategoryID = category.WidgetCategoryID;

            // Save new widget to database
            WidgetInfoProvider.SetWidgetInfo(wi);
        }
```

The following sample code shows how you can delete an existing widget from the system:

[C#]

```
using CMS.PortalEngine;

...

        // Get widget from database
        WidgetInfo wi = WidgetInfoProvider.GetWidgetInfo("AbuseReport");

        // If widget exists
        if (wi != null)
        {
            // Delete from database
            WidgetInfoProvider.DeleteWidgetInfo(wi.WidgetID);
        }
```

7.19.5.4 Setting widget permissions

The following sample code shows how you can assign a role and permission to a widget:

[C#]

```
using CMS.PortalEngine;
using CMS.SiteProvider;
using CMS.CMSHelper;

...

        string permissionName = "AllowedFor";
        string resourceName = "Widgets";
        string className = null;
        string roleName = "TestRole";
        int siteID = CMSContext.CurrentSiteID;

        // Get role, widget and permission object
        RoleInfo ri = RoleInfoProvider.GetRoleInfo(roleName, siteID);
        WidgetInfo wi = WidgetInfoProvider.GetWidgetInfo("AbuseReport");
        PermissionNameInfo pni = PermissionNameInfoProvider.GetPermissionNameInfo
(permissionName, resourceName, className);

        // If all exist
        if ((ri != null) && (wi != null) && (pni != null))
        {
            // Add role to widget
            WidgetRoleInfoProvider.AddRoleToWidget(ri.RoleID, wi.WidgetID, pni.Permi
        }
```

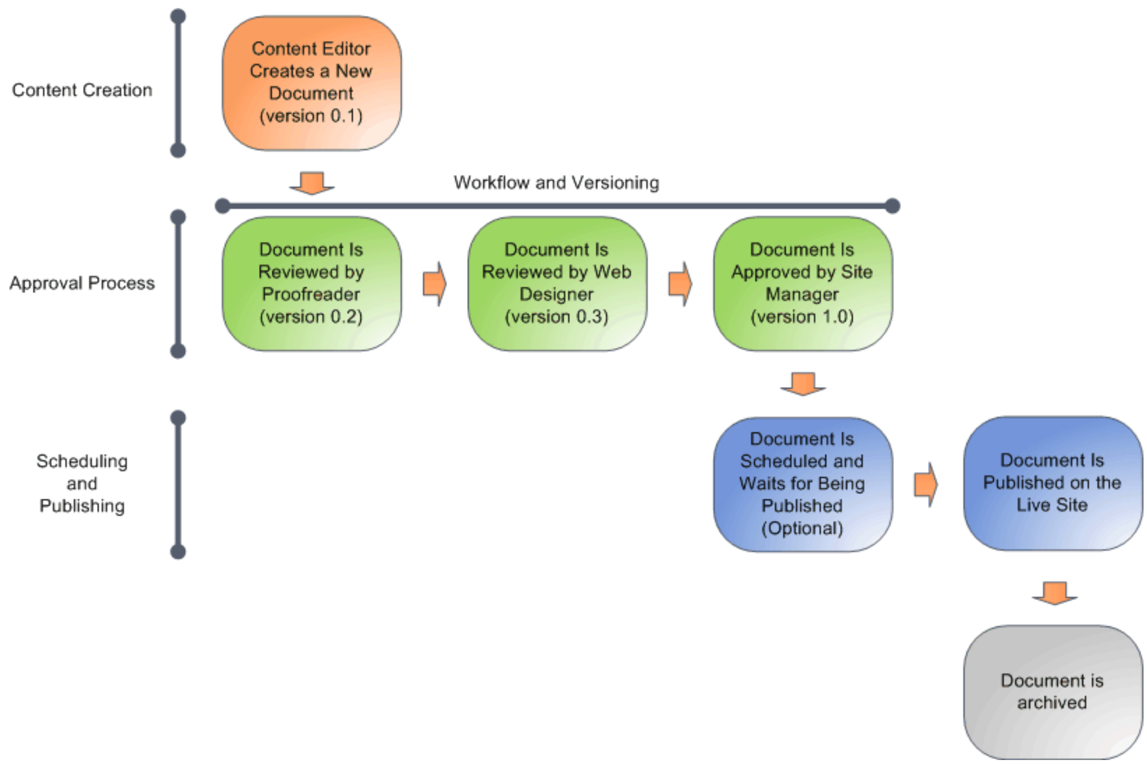
7.20 Workflow and versioning

7.20.1 Overview

Kentico CMS allows you to use workflow for all documents, including uploaded files. It includes the following features:

- The **workflow** support allows you to organize the process of content creation, updates and publishing on your website.
- The **versioning** support is tightly bound with workflow and it allows you to store (audit), view and roll-back previous versions of the content.
- It is also possible to use **versioning without workflow**, which creates a new version whenever a document is modified, while the whole editing process remains as if the document was using no workflow at all.
- The **content locking (check-in/check-out)** support allows you to avoid concurrent modifications of the same document by multiple users.
- The **preview** support allows you to see the content in the context of the site before it's published. This feature is only available for documents that use workflow - you can preview the documents before switching them to the following step.
- The **archiving** support allows you to archive a document. Such a document is no longer displayed on the website, but it's kept in the content tree and can be re-published at any time later.

The following figure shows an example of document life cycle with a workflow:



The workflow process consists of any number of custom, sub-sequent steps. Only members of authorized roles are allowed to modify and approve/reject a document in a particular step. The document can be switched to the following step by any authorized role members.

As you can see, the workflow process can be used to ensure quality of content and design. In the following topics, you will learn how to define and apply workflow processes for your documents.

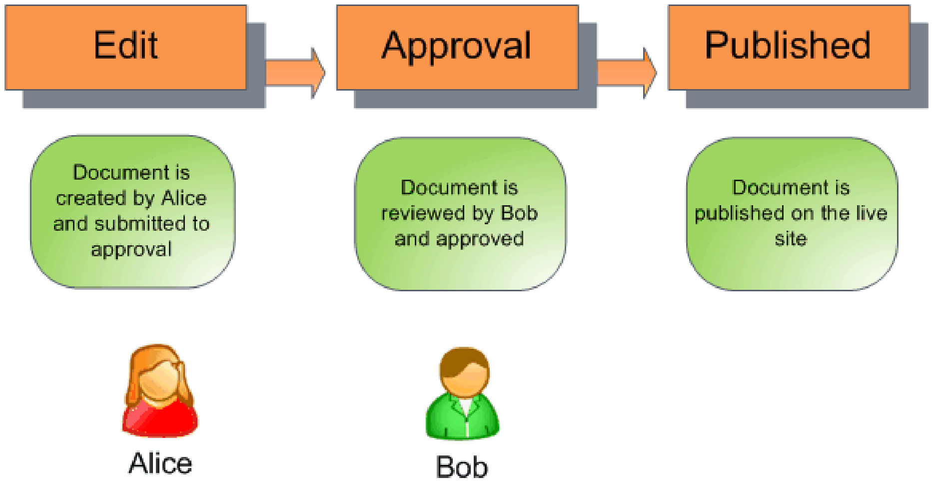
7.20.2 Defining a workflow

Workflow settings can be managed in **Site Manager -> Development -> Workflows**.

In this topic, we will create a sample workflow which will be used in the following topics. The workflow can be described as follows:

- Any editor will be able to create a news item in the **/News** section of the website.
- Then, the news item must be approved by one of the members of the **PR Managers** role. Users authorized to approve the document in the given step are also authorized to modify the document in the step.
- Approved news item is published on the website.

This a simple example of a typical real-world workflow containing custom steps. It is also possible to configure a workflow so that it contains no custom steps and is used just for versioning purposes. Configuration of such workflow is described in the Versioning without workflow topic.



Creating testing users and roles

Before we configure the workflow, we will create testing users and testing roles whose members the users will be.

1. Go to **Site Manager -> Administration -> Users** and click **New user ()**. Create a new user with the following details:

- **User name**: Alice
- **Full name**: Alice Murphy
- **E-mail**: <your e-mail>
- **Enabled**: yes
- **Is editor**: yes

Click **OK**. Go to the **Sites** tab and assign the user to your current website. Go to the **Roles** tab and assign the user to the **CMS Editors** role of the given website.

2. Create another user with the following details:

- **User name**: Bob
- **Full name**: Bob Johnson
- **E-mail**: <your e-mail **-** you may want to use a different e-mail for testing purposes>
- **Enabled**: yes
- **Is editor**: yes

Click **OK**. Go to the **Sites** tab and assign the user to your current website. Go to the **Roles** tab and assign the user to the **CMS Editors** role of the given website.

3. Go to **Site Manager -> Administration -> Roles**, choose the appropriate site from the top drop-down-list and click **New role ()**. Enter the following details:

- **Role display name**: PR Managers
- **Role code name**: PRManagers

Switch to the **Users** tab and add user *Bob Johnson* to this role.

As you can see, Alice is the news editor (because of the *CMS Editors* role) and Bob is the PR Manager who will approve the news (*PR Managers* role) and can also edit the news (*CMS Editors* role).

Please note

The *PR Managers* role has no permissions configured for the purpose of this example. This means that if a user was a member of just this single role, they would not be able to perform a full range of tasks such as creating, editing or deleting documents, etc. In your real website, you would typically need to go to **Site Manager -> Administration -> Permissions** and grant appropriate permissions to the role, especially those in the **Modules -> CMS Content** permission matrix.

Please see the Development -> Membership, permissions and security -> Permissions chapter for detailed information on how to do this.

Defining a new workflow

With the testing users and roles defined, we may start creating the actual workflow.

4. Go to **Site Manager -> Development -> Workflows**, click **New workflow ()** and enter the following values:

- **Display name**: News approval
- **Code name**: NewsApproval
- **Automatically publish changes**: disabled
- **Use check-in/check-out**: No

Click **OK** to create the workflow.

Defining custom workflow steps

If you switch to the **Steps** tab, you should see that each workflow has **three default steps**:

- **Edit** - the document was created or modified after it was published.
- **Published** - the document is published on the live site. When you edit such a document, it's automatically moved to the **Edit** step.
- **Archived** - the document is not published on the live site, but it remains in the content repository.

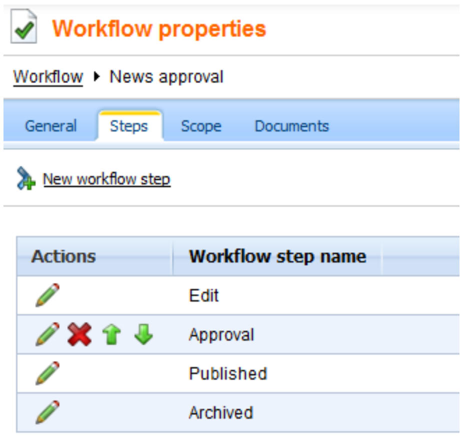
We will add one extra custom step to the workflow.

5. Switch to the **Steps** tab and click **Add workflow step ()**. Enter the following details in the displayed dialog:

- **Display name**: Approval
- **Code name**: Approval

Click **OK**. In the workflow step editing interface, switch to the **Roles** tab. Choose the appropriate website in the drop-down list and add the **PR Managers** role to the list. Now the *PR Managers* are authorized to edit, approve or reject documents in this step.

When you view the list of workflow steps now, it looks as in the screenshot below. As you can see, the custom steps are placed between the **Edit** and **Published** steps. If there is more than one step, you can change their order using the up ( ) and down ( ) arrows.



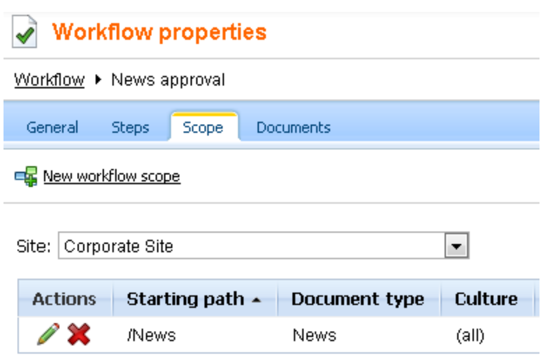
Defining a workflow scope

Workflow scopes define which documents should the workflow be applied to.

6. Switch to the **Scopes** tab, choose the appropriate website from the drop-down list and click **New workflow scope ( )**. Enter the following details:

- **Starting alias path**: */news* (this is the section of the website for which the workflow will be applied)
- **Document type**: *News*
- **Culture**: *(all)*

Click **OK**.



You have configured the workflow process for the news documents. In the Using workflow topic, you will learn how the workflow works in our sample scenario.

Workflow's documents overview

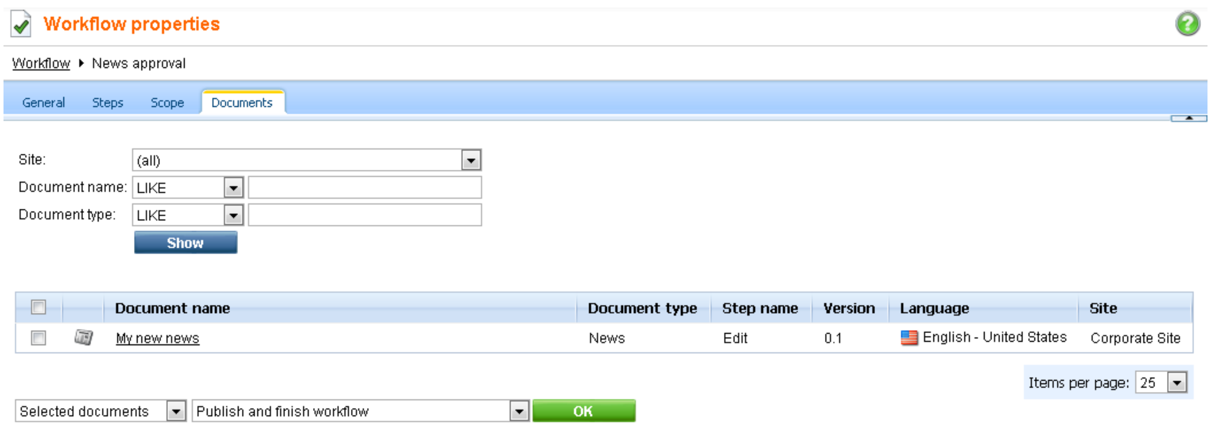
On the **Documents** tab of a workflow's editing interface, you can see a list of documents that are using the workflow. Clicking a document in the list opens it for editing in **CMS Desk -> Edit**.

The following actions can be performed with documents in the list. These actions are also useful if you want to **finish the documents' workflow**:

- **Publish and finish workflow** - the new version of the document will be approved and published
- **Remove workflow and keep currently published data** - the original version of the document remains published and all the changes will be discarded

Based on the selection in the first drop-down list, these actions can be performed for:

- **All documents** - the action will be performed with all documents
- **Selected documents** - the action will be performed with documents selected by the check-boxes ( )



7.20.3 Using workflow

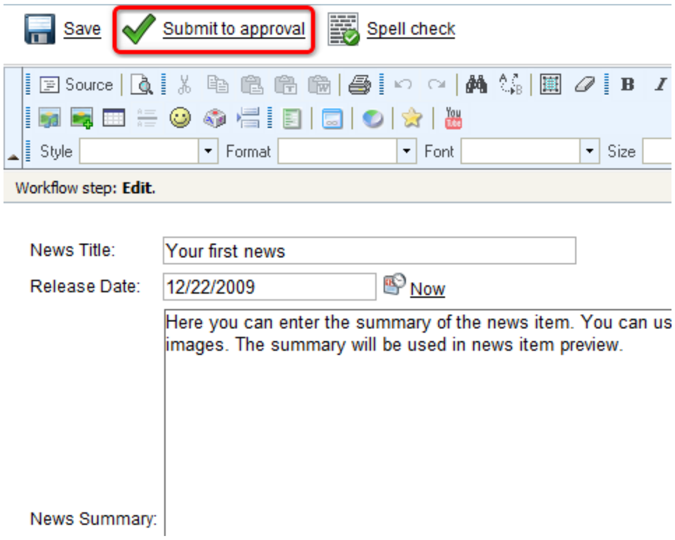
This example explains how the workflow process works from the perspective of content editors. It requires the system to be configured as described in Defining a workflow.

Creating a new document

1. Sign in to **CMS Desk** as **Alice** and create a news item in the **/News** section:

- **News title**: My first news item
- **Release date**: <click **Now** to get the current date>
- **News summary**: Some summary
- **News text**: Some text
- **Publish from-to**: leave the fields blank

Click **Save**. After the document is saved, you can see that there is the **Submit to approval** action displayed in the document menu:



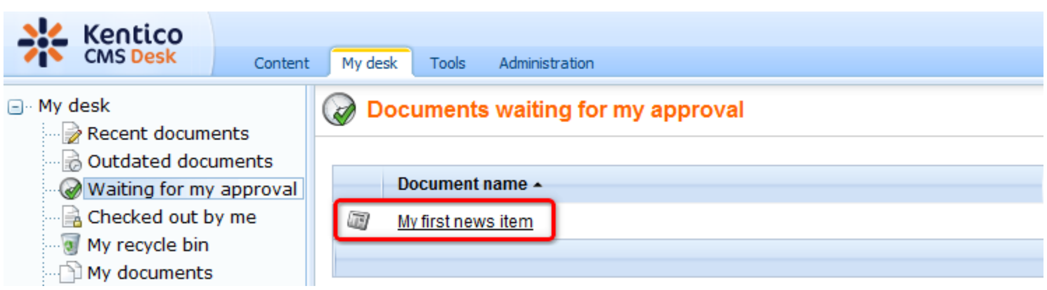
2. Before we submit the document to approval, we will preview it for a final check. Click **Preview** in the main toolbar and you will see exactly what the document will look like on the live site. After doing so, go back to the editing form and click the **Submit to approval** button.

The document has been moved to the **Approval** step and it is now waiting for approval by some member of the *PR Managers* role and Alice can no longer edit it. Also, a notification e-mail is sent to all members of the PR Managers role.

Approving the document

With the document submitted to approval, we can switch to the PR Manager's perspective and approve it.

3. Sign out of **CMS Desk** and sign in again as **Bob**. Go to **My desk -> Waiting for my approval**. You should see the new document listed as in the screenshot below.



4. Click the document, which will open it for editing in **CMS Desk -> Content -> Edit**. You can now preview and optionally edit it.

If you were not satisfied with it and wanted Alice to re-work it, you could click **Reject** to return it back to the **Edit** step. For the purpose of this example, we will approve it by clicking the **Publish** button. Once you approve the document, it gets published on the live site. A notification e-mail is sent to Alice so that she knows the document has been published if configured as described in E-mail notification in workflow process.

Tip 1: Approval/rejection comments

When you're approving or rejecting the document, you can go to **Properties -> Workflow**, add additional comments and click **Approve**.

Tip 2: "Documents waiting for my approval" and "Recent documents"

Each user can find a list of documents waiting for their approval in **My desk -> Documents waiting for my approval**.

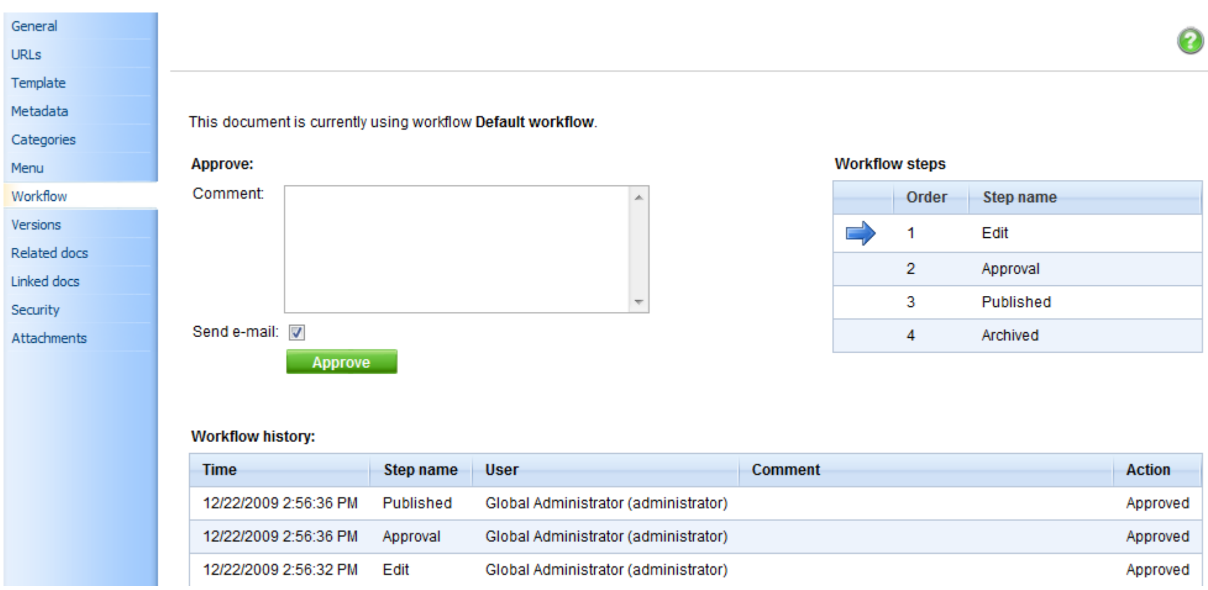
Similarly, the user who submitted the document can see the document status in **My desk -> Recent documents**.

Archiving the document

If you want to archive the document so that it's no longer displayed on the live site, but it remains in the Kentico CMS database, you can go to **Properties -> Workflow** dialog of the published document and click the **Archive** button.

Workflow history and workflow comments

Beside the action buttons for approving/rejecting the document on the editing pages, you can also use the advanced interface in **Properties -> Workflow dialog**:



Here you can also track the workflow history of the document and see the current workflow step in the workflow.

Workflow and permissions

Once you enable workflow, the user needs to be authorized to both edit the document and must be authorized to manage the document in the give workflow step.

The users with permission "Manage workflow" can approve/reject any document in any workflow step, even if they are not authorized for the given workflow step. This permission can be set in the **Site Manager -> Administration -> Permissions** section, in the permission matrix called **Module: CMS Content**.

7.20.4 E-mail notification in workflow process

The workflow process automatically notifies the members of the authorized roles that there's a document waiting for their approval. It also sends a notification to the document author (the user who last edited the document in the Edit step and sent it to the next workflow step) when the document status changes. Lastly, a notification is also sent to all users who have the **Manage workflow** permission for all content .

Please note: notification e-mails are not sent among users who have the same e-mail address, i.e. when user1 and user2 have the same address and user1 sends a document for user2's approval, no notification e-mail is sent

You can turn off e-mail notifications in the workflow process in **Site Manager -> Settings -> Content Management**. Here you can set the following values:

- **Send workflow e-mails** - indicates if workflow e-mails should be sent.
- **Send workflow e-mails from** - the e-mail address from which the e-mails will be sent.

Tip: Workflow notification e-mail templates

You can customize the workflow notification e-mails in **Site Manager -> Development -> E-mail templates**. The workflow uses the following e-mail templates:

- Document approved
- Document archived
- Document published
- Document ready for approval
- Document rejected

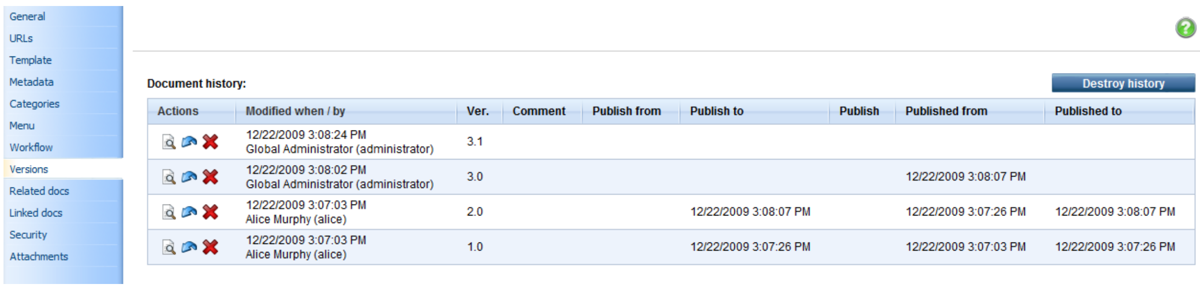
You can use the following macro expressions in the workflow notification templates:

- **{%applicationurl%}** - the URL of the Kentico CMS web project on the server (including protocol and domain name)
- **{%nodeid%}** - NodeID of the current document
- **{%documentname%}** - document name
- **{%approvedby%}** - name of the user who approved/rejected/archived the document or submitted the document to approval
- **{%approvedwhen%}** - date and time of approval/rejection/archivation

- **{%originalstep%}** - original step before approval/rejection/archivation
- **{%currentstep%}** - new step after approval/rejection/archivation
- **{%comment%}** - comment added by the user who changed the status of the document

7.20.5 Versioning and rollback

Versioning allows you to store all versions of a document which uses workflow. You can view the versions of a document in **CMS Desk -> Content -> Properties -> Versions**.



You can perform the following actions with the listed versions:

- **View -** displays an overview of the document's content with the possibility of side-by-side comparison of versions
- **Rollback -** rolls back any changes made since the particular version of the document
- **Delete** - deletes the old version

You can use the **Destroy history** button to clear the history and restart version numbering. The **Destroy** permission for this document must be granted to the user.

The history also shows when the particular version has been published and when it was replaced with a new version.

The length of the version history can be configured in **Site Manager -> Settings -> Content management -> Version history length**.

Automatic version numbering

Kentico CMS supports automatic version numbering:

- If you use workflow without content locking, automatic version numbering is used by default.
- If you use content locking, this option is optional and can be configured in **Site Manager -> Settings -> Content management -> Use automatic version numbering**.

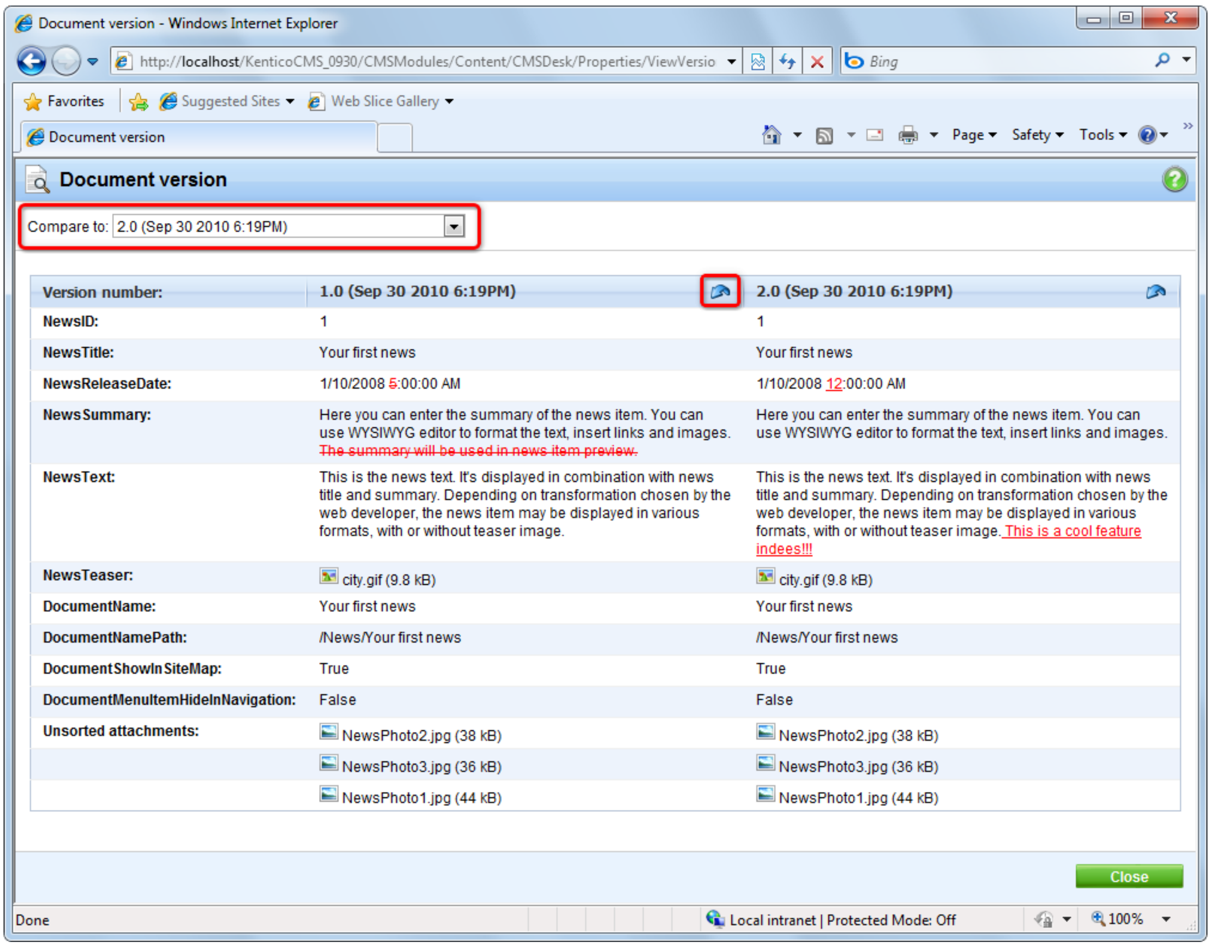
The automatic version numbering works like in the following example:

- 0.1 - first version of the document when it's created
- 0.2 - second modification of the document
- 1.0 - first published version of the document
- 1.1 - first modification of the published document
- 2.0 - second published version of the document

Side-by-side comparison of versions

If you view click the **View** ( ) icon next to a version, the pop-up window allows you to view a side-by-side comparison with other document versions.

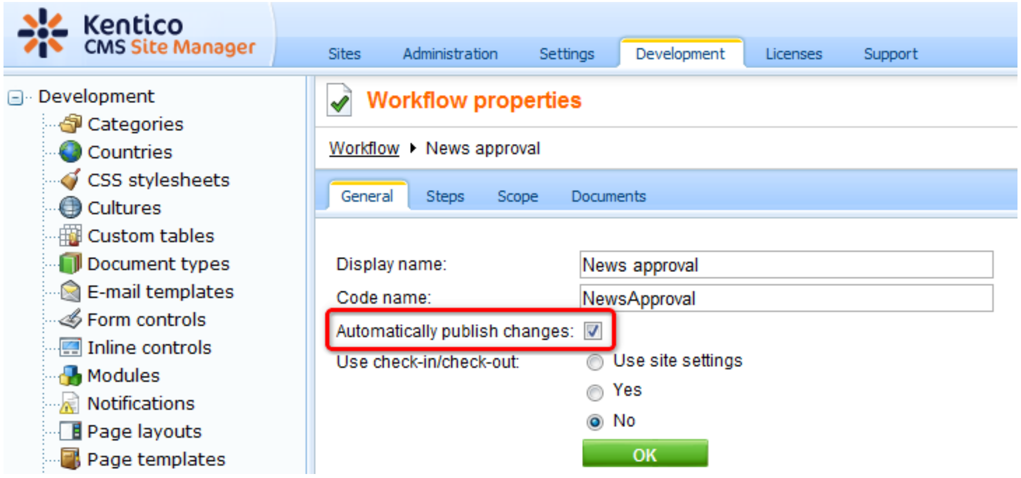
You only need to select the appropriate version in the **Compare to** drop-down list. Content of the two versions will be displayed side-by-side with the changes highlighted in red font. You can roll back the document to a particular displayed version by clicking the **Rollback** ( ) icon in the top-right column of the version's column.



7.20.6 Versioning without workflow

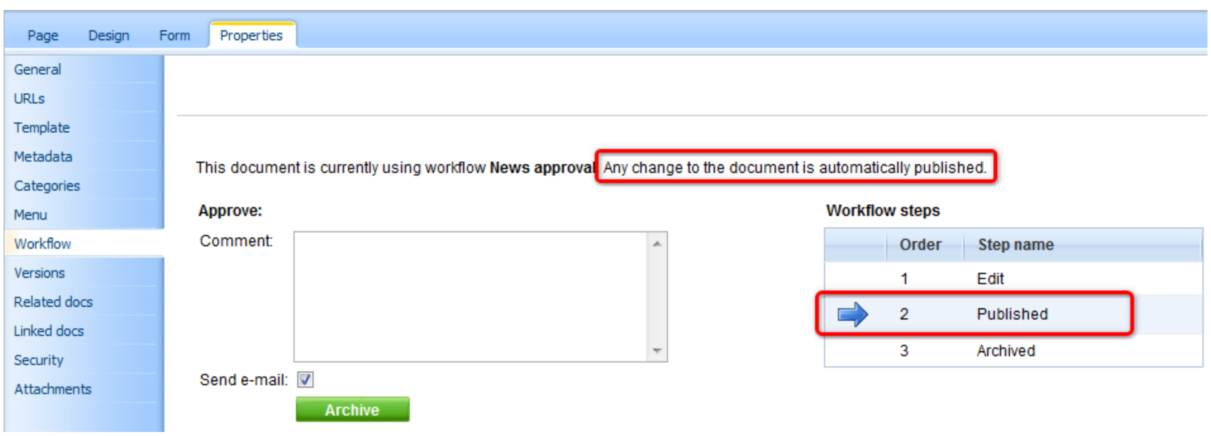
It is possible to use versioning without actually using any workflow steps. In other words, it is possible to have a new document version created whenever a document is modified and saved.

To enable versioning without workflow, you only need to enable the **Automatically publish changes** option either when creating a new workflow, or when editing ( ) a workflow on its **General** tab. With this option enabled, custom steps can not be created on the **Steps** tab. You only need to define appropriate scope(s) so that the workflow affects the required documents.



From the end-user point of view, editing a document using versioning without workflow is the same as if the document was using no workflow at all. The **Submit to approval**, **Approve** and **Reject** buttons are not displayed on the document editing form - there is just the standard **Save** button present.

On the **Properties -> Workflow** tab, you can recognize a document using versioning without workflow by the *"Any change to the document is automatically published."* sentence, as highlighted below. You may also notice that the document remains in the **Published** step all the time until it gets archived.



7.20.7 Content locking

Content locking allows content editors to lock a document for editing, so that other editors cannot modify the document at the same time.

Content locking configuration

You can enable content locking for versioned documents at two levels:

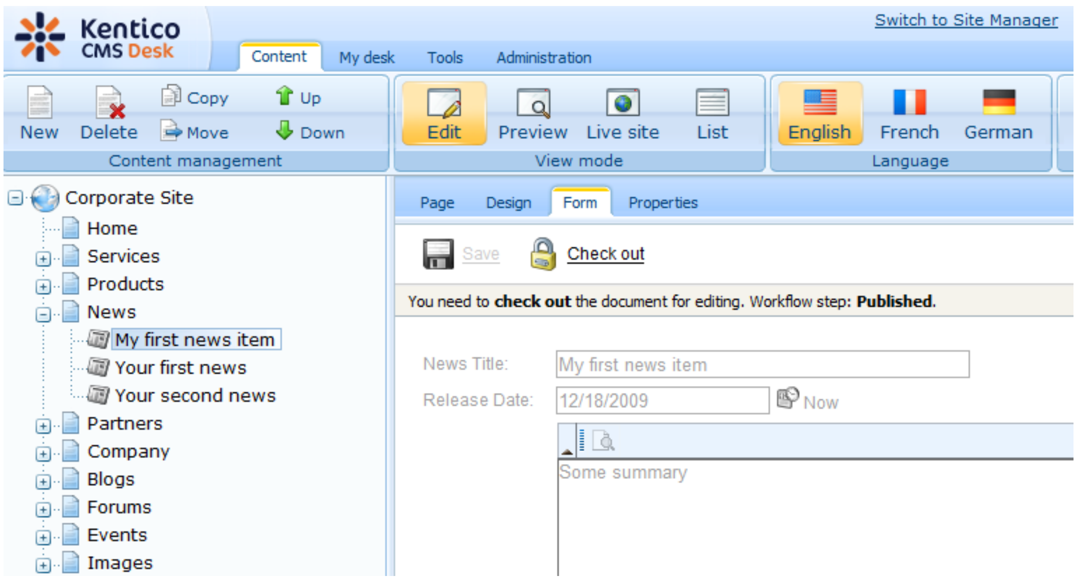
- globally in **Site Manager -> Settings -> Content management -> Use check-in/check-out**
- separately for each particular workflow, in **Site Manager -> Development -> Workflows -> edit ( ) a workflow -> General**; here, you can configure the **Use check-in/check-out** option to either inherit from the global settings, or set explicitly if content locking should be used with this workflow

Example

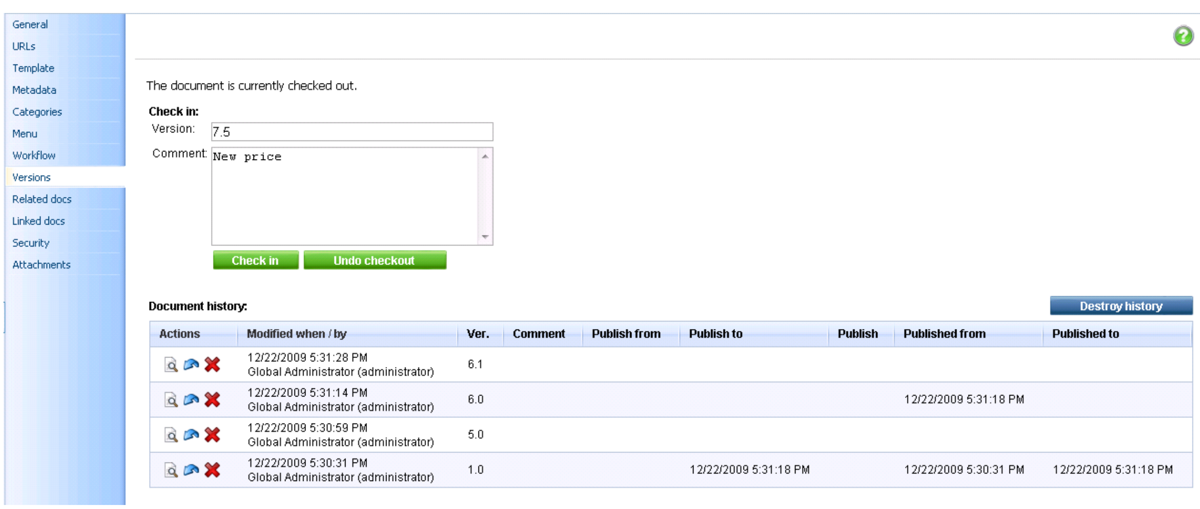
Here's an example of operations in the document life cycle when content locking is used:

1. User Alice creates a news item. The news item is automatically checked out (locked) for her. The other users cannot edit it.
2. User Alice finishes the changes and checks in the document.
3. Now the other users can modify the document as well. As the document is checked in, Alice can also submit it to approval.

When you try to edit a checked in document, you will get a disabled editing form with **Check out** button:



You can also use the **Check in/Check out** buttons in **Properties -> Versions dialog**. This dialog allows you to specify custom version numbers and comments for each version when you check in the document:



Checking-in any document

Users to whom the **Check in any document** permission was granted can check-in any document, even if they haven't checked-out the document. This permission can be set in the **Site Manager -> Administration -> Permissions** section, in the permission matrix called **Modules -> CMS Content**.
However, this check-in can only be performed from the **Properties -> Versions** tab of the selected document, by clicking either the **Check in** or **Undo check-out** buttons.

7.20.8 Managing workflow using .NET code (API)

For details on versioning and workflow implementation and their scripting using .NET API, please see chapters Workflow internals and Versioning internals.

Part VIII

Modules

8 Modules

8.1 Overview

This section provides reference on in-box modules in Kentico CMS. Information on particular modules can be found in the sub-categories.

You can also learn how to develop a simple module and integrate it in Kentico CMS's user interface in this example.

Please note: not all modules might be available in your installation because some modules are available only with certain licenses

8.2 Developing custom modules

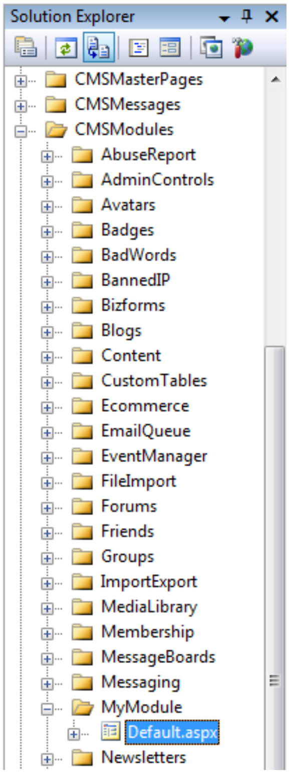
This chapter describes how you can create a custom module integrated into Kentico CMS administration interface.

You will learn how to create a simple module with a single button that displays the current date and time. The procedure can be used for adding any kind of custom module that you develop. You will also learn how to control access to the module's functions using permissions and integration of this module into the UI using UI personalization.

1. Open Kentico CMS web project in Visual Studio. You can do that either using the *WebProject.sln* file in your website root or using the **File -> Open -> website** menu in Visual Studio.

2. Create a new folder *MyModule* under the *CMSModules* folder.

3. Create a new page *default.aspx* under the *CMSModules/MyModule* folder:



4. Switch to code behind of the page and change the following line:

[C#]

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

to:

[C#]

```
public partial class CMSModules_MyModule_Default : CMSToolsPage
```

It ensures that the module can be used only by users with access to Kentico CMS Desk.

5. Add the following code at the beginning of the page:

[C#]

```
using CMS.CMSHelper;
using CMS.UIControls;
```

6. Add the following code to the *Page_Load* method:

[C#]

```
if (!CMSContext.CurrentUser.IsAuthorizedPerResource("cms.mymodule", "read"))
{
    RedirectToAccessDenied("cms.mymodule", "Read");
}
```

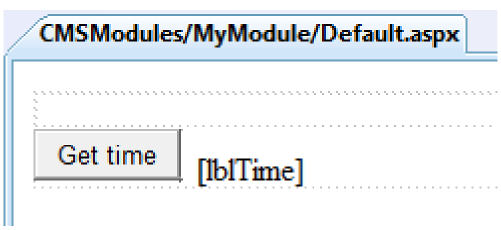
It checks if the current user has the **Read** permission for module *myprojects.mymodule*.

7. Switch to the **Design** mode and add a **Button** control on the page. Set its properties:

- **ID**: btnGetTime
- **Text**: Get time

8. Add a **Label** control on the page, next to the button. Set its properties:

- **ID**: lblTime
- **Text**: (clear the value)



9. Double-click the button and add the following code inside the **btnGetTime_Click** method:

[C#]

```
if (CMSContext.CurrentUser.IsAuthorizedPerResource("cms.mymodule", "gettime"))
{
    lblTime.Text = DateTime.Now.ToString();
}
else
{
    lblTime.Text = "You're not authorized to get the current date and time.";
}
```

It checks if the current user has *gettime* permission for a module named *cms.mymodule* and if so, it displays the current date and time.

10. Run the project and sign in to **Site Manager**. Go to **Development -> Modules** and click **New module**. Enter the following values:

- **Module display name**: My module
- **Module code name**: cms.mymodule; for the UI element to be site-related, it is important to enter the value in the *cms.<elementname>* format, where *<elementname>* is the Code name of the UI element created in step 16; see this topic for more details

Click **OK**. The module was registered in the system.

Please note: in the system, modules are represented by *ResourceInfo* and *ResourceInfoProvider*

11. Go to the **Permission names** tab and click **New permission**. Enter the following values:

- **Permission display name**: Read
- **Permission code name**: read
- **Display in matrix**: enable

Click **OK**.

12. Add another permission:

- **Permission display name**: Get time
- **Permission code name**: gettime
- **Display in matrix**: enabled

Click **OK**.

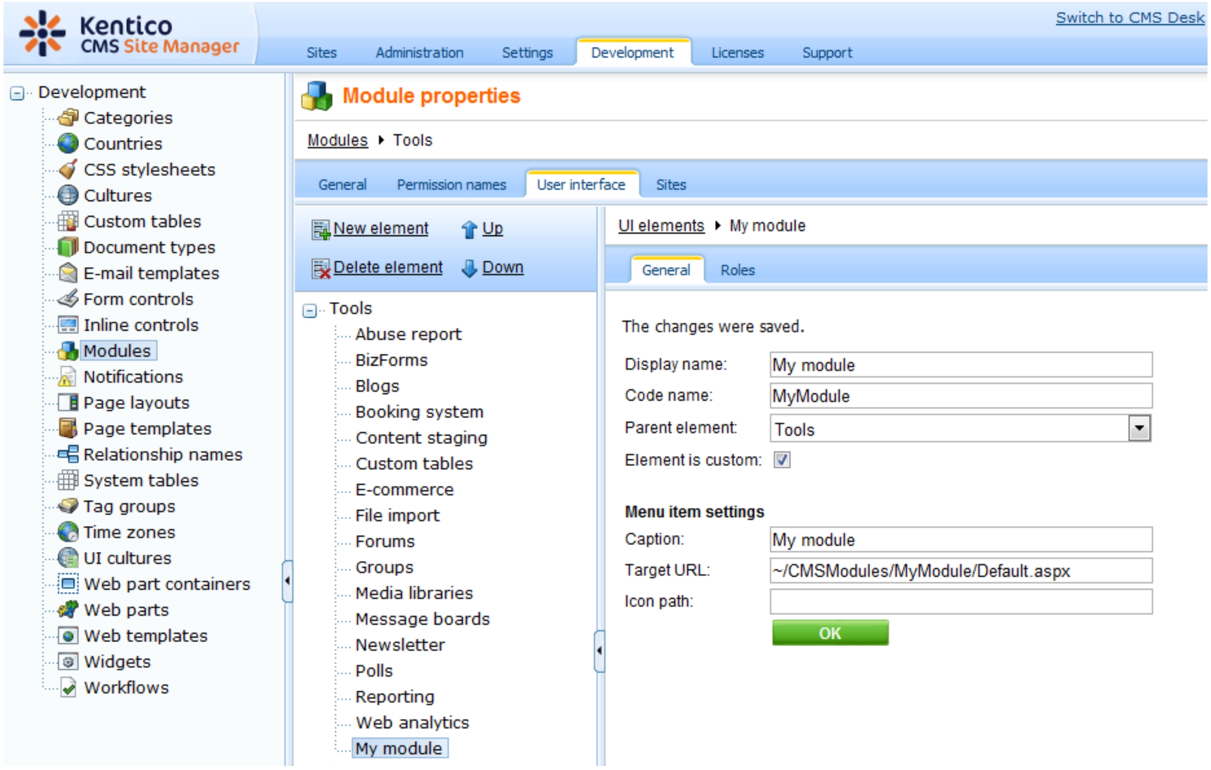
13. Go to the **Sites** tab and enable the new module for appropriate sites.

Now that the module is registered, we need to display it in the user interface. We will do it by registering a new UI element as described here. For the purposes of this example, we will place it into the **Tools** menu. The module can also be placed into other sections of the UI as listed here.

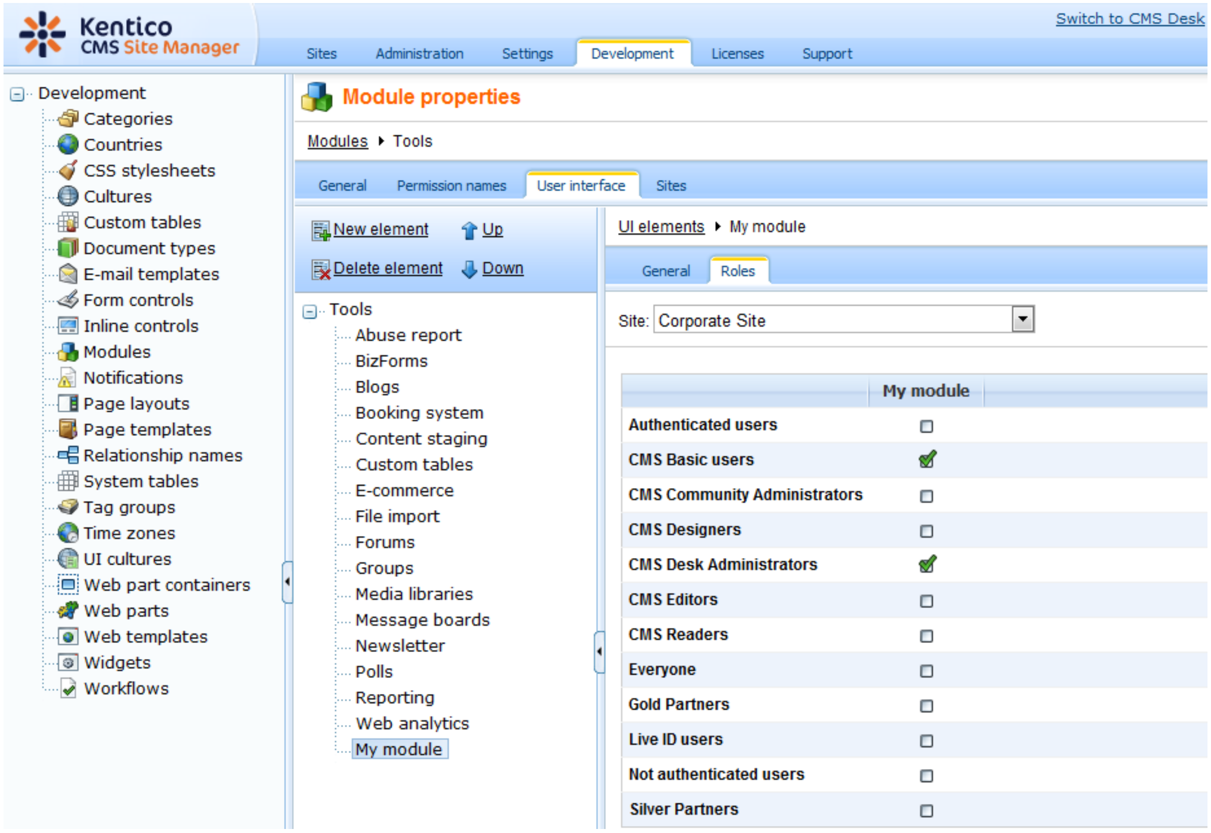
14. Go back to **Development -> Modules** and edit () the **Tools** module. Switch to its **User interface** tab, select the root of the tree (Tools) and click **New element**. Enter the following details:

- **Display name**: My module
- **Code name**: MyModule
- **Element is custom**: enabled
- **Caption**: My module
- **Target URL**: ~/CMSModules/MyModule/Default.aspx

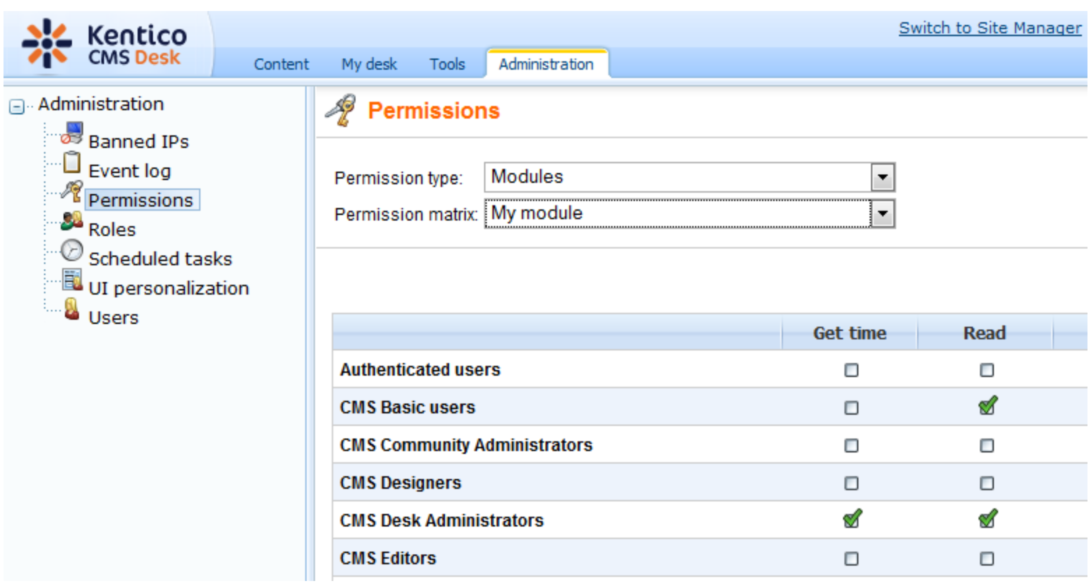
Click **OK**.



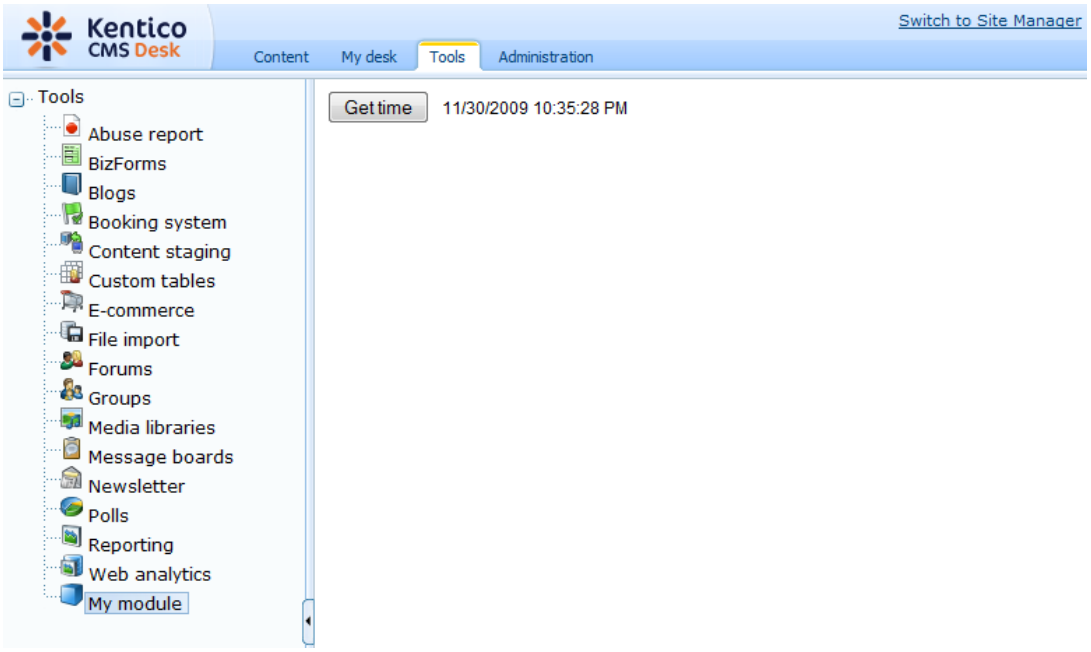
15. We will use UI personalization to enable access to this module only to members of certain roles. For this to work, UI personalization needs to be enabled. With the new UI element still selected, switch to the **Roles** tab and enable the element for members of the **CMS Basic users** and **CMS Desk Administrators** roles.



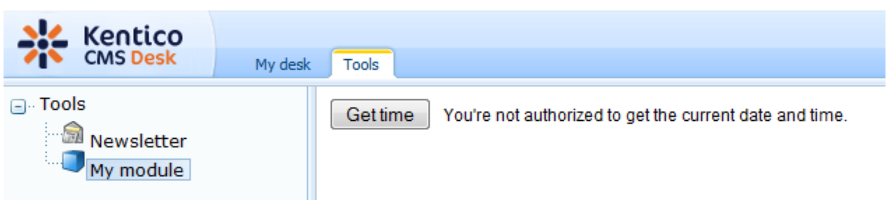
16. Go to **CMS Desk -> Administration -> Permissions** and choose **Permission type**: Modules with **Permission matrix**: My module. Grant the *Read* permission to the **CMS Basic users** role and both the *Get time* and the *Read* permissions to the **CMS Desk Administrators** role:



17. Now go to **CMS Desk -> Tools** logged in as *administrator* (with blank password). You can see the **My module** item at the bottom of the left menu. Click it and you will see your form created in steps 3-9. When you click the button, current date and time gets displayed.



18. Now try logging in as *andy* (with blank password) and go back to **CMS Desk -> Tools**. Andy is a member of the **CMS Basic users** role. In step 18, you configured that members of this role have only the *Read* permission necessary for the module to be displayed (ensured in step 6), but not the *Get time* permission necessary for the button functionality. Try clicking it, you will see the access denied message that you defined in step 9.



Exporting custom modules

The following folders will be included in the export package when the custom module is selected for export. It is therefore recommended that your modules data is stored within these folders:

- ~/App_Code/CMSModules/<module_name>
- ~/App_Data/CMSModules/<module_name>
- ~/CMSModules/<module_name>

The <module_name> value needs to be the same as the code name of the module in the administration interface, so for example for a module named CMS.Test, the folders would be:

- ~/App_Code/CMSModules/CMS_Test

- ~/App_Data/CMSModules/CMS_Test
- ~/CMSModules/CMS_Test

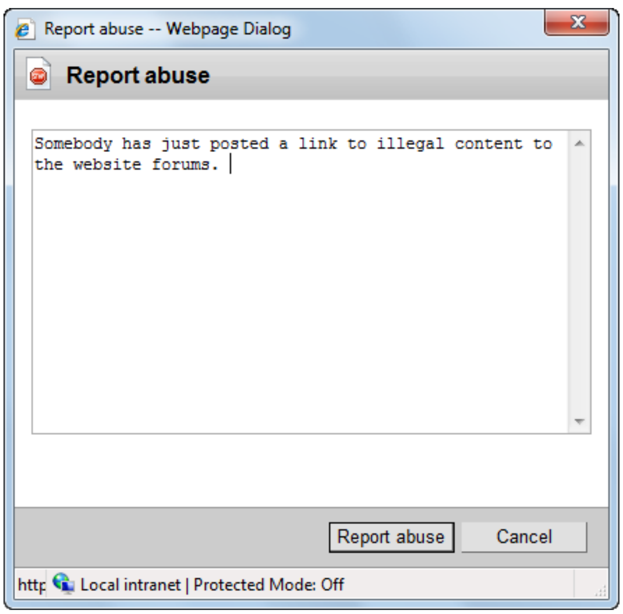
8.3 Abuse report

8.3.1 Overview

The Abuse report module enables site visitors to report various forms of website abuse. This can be anything from flame-type forum posts, rude comments on blog posts or message boards, spamming, links to illegal content or any other form of website abuse coming from user input.

The whole functionality is based on website users reporting these forms of website abuse to website administrators via one of the provided web parts - **Abuse report** and **In-line abuse report**. Both web parts are described in the Available web parts chapter. The Using the In-line abuse report web part in transformations chapter demonstrates how the **In-line abuse report** web part can be used in transformations to provide the abuse report functionality together with dynamically loaded content.

Abuse report functionality is also embedded in web parts of the Forums and Message boards modules. Advice on how to use the embedded functionality can be found in the Integration with other modules chapter.



Website administrators can read and manage submitted reports in **CMS Desk -> Tools -> Abuse report**. This is where a list of submitted reports is displayed, with the possibility to get redirected to the source of reported abuse and take an appropriate action. All abuse report management possibilities provided in this part of the administration interface are described in the Abuse reports management chapter. A user who wants to manage abuse reports must be in a role granted with appropriate permissions for the module - please consult the Security chapter for more details on the module's permissions.

The Abuse report internals and API sub-chapter provides information about the database tables and classes used by the module, as well as examples of how abuse reports can be managed using Kentico CMS API.

The Abuse report module is just one of the modules which can help you deal with unwanted user behavior on your site. If you want to keep your site free of rude language or simply filter user input based on contained keywords, the Bad words module may come in handy. If you are facing a problem with notorious spammers or have another reason why an IP address should be prevented from accessing your site, you can deny access to a particular IP address using the Banned IPs module.

8.3.2 Available web parts

This page gives just a brief overview of the Abuse report web parts. Detailed descriptions of each web part and its properties can be found in Kentico CMS Web Parts Reference.

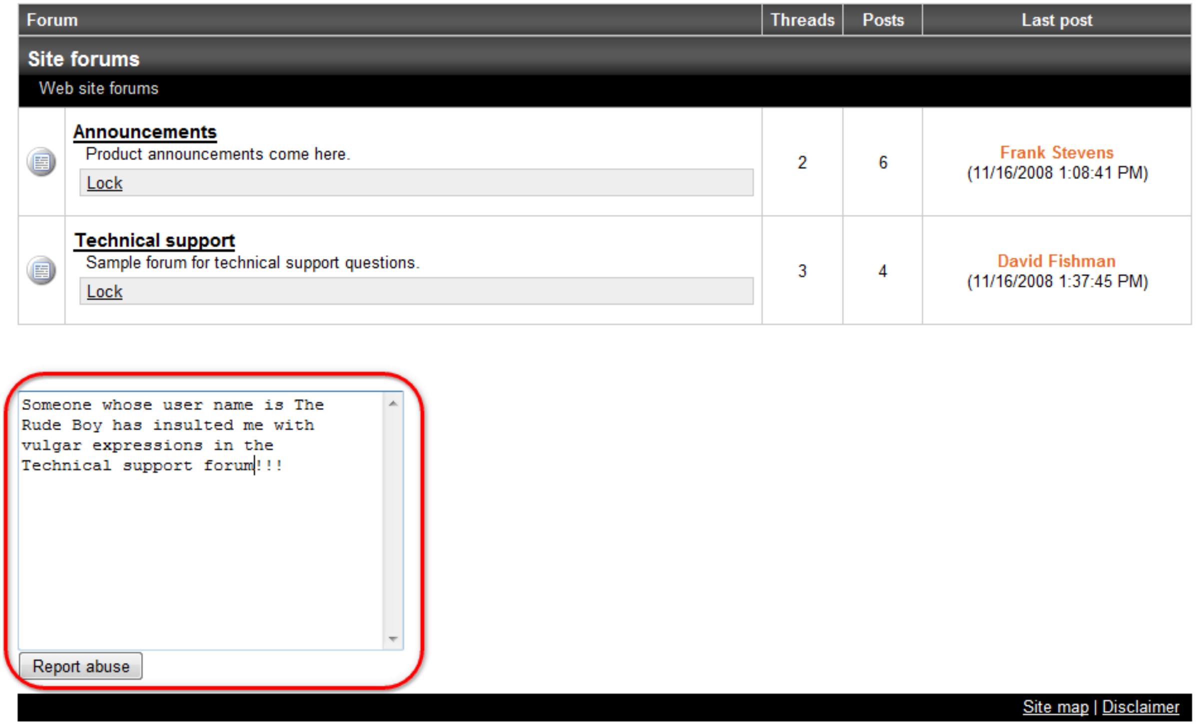
Abuse report

The **Abuse report** web part gives site visitors the possibility to report website abuse such as using indecent expressions, offensive language, etc. In the web part selection dialog, you can find it under the **Abuse report** folder. It can be placed into any web part zone on any page of your website, while you will typically add it to pages where user input is possible, e.g. pages with forums, message boards, blogs, user contributions, etc.

You only have to set the following two specific properties in order for the web part to work properly:

Confirmation text	Text message that will be displayed to the site visitor after submitting the abuse report.
Title of abuse report	Title of the report that will be displayed in the list of abuse reports in **CMS Desk -> Tools -> Abuse report**. Using this property, you can distinguish reports submitted at different pages of your website from each other.

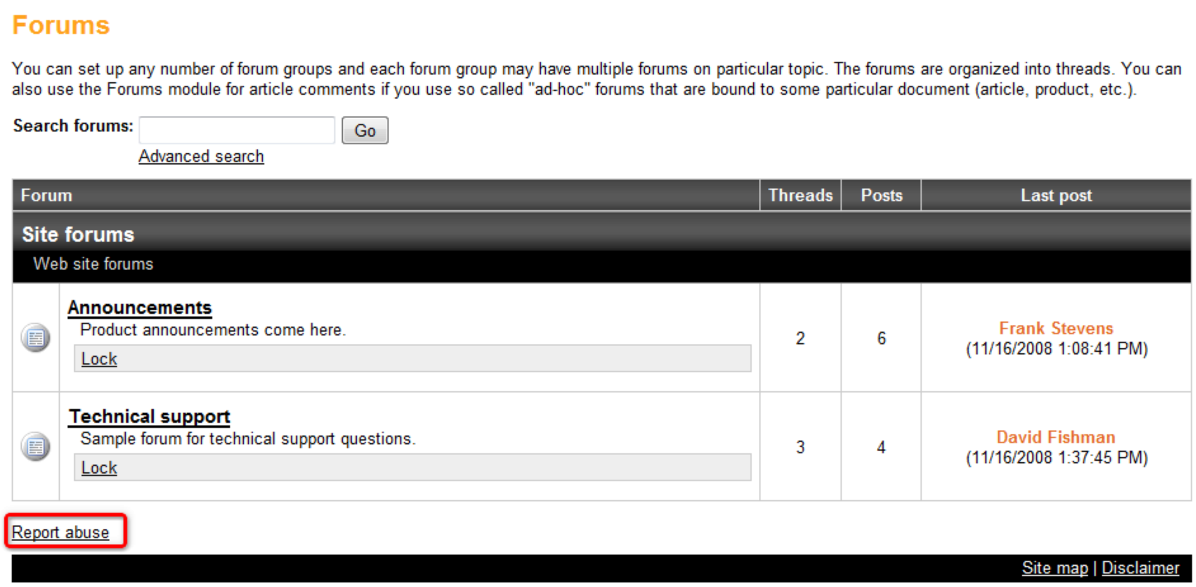
In the screenshot below, you can see the web part added below the forum group on the **/Forums** page of the sample Corporate Site. When a user types in some report text and clicks the **Report abuse** button, a report is logged in **CMS Desk -> Tools -> Abuse report**. Read the Abuse reports management topic for information on how reports can be managed in this section of CMS Desk.



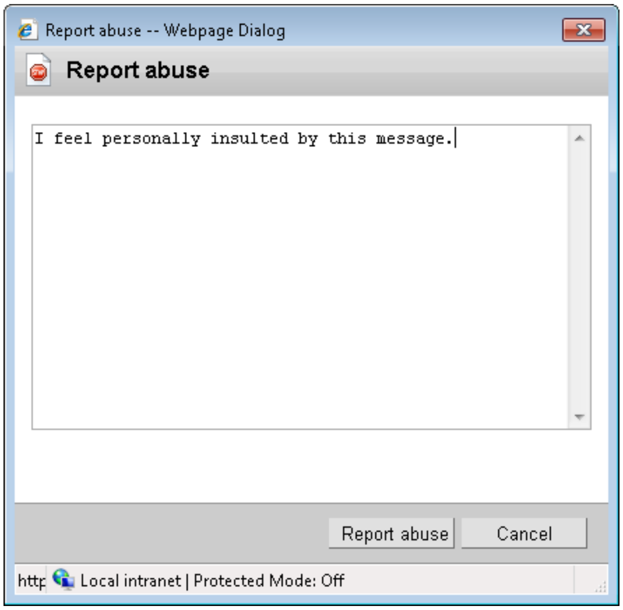
In-line abuse report

If you want to use only a tiny clickable link instead of the whole **Abuse report** web part, you can use the **In-line abuse report** web part. The web part appears only as the *Report abuse* link, as you can see in the screenshot below. Properties of the two web parts are identical.

The **In-line abuse report** web part can also be used in transformations, as explained in the Using the In-line abuse report web part in transformations chapter.



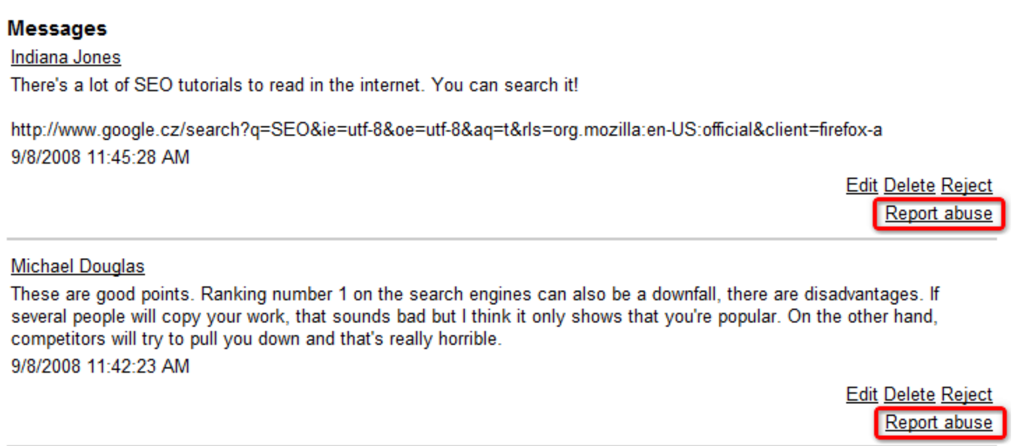
When a site visitor clicks the link, a dialog pops-up, letting the site visitors report abuse to the site administrators. Reports submitted via this web part are also logged in **CMS Desk -> Tools -> Abuse report** and can be managed there as described in the Abuse reports management topic.



8.3.3 Using the In-line abuse report web part in transformations

The Inline abuse report web part appears as a link with the *Report abuse* text. After clicking the link, a dialog appears, letting the site visitor send abuse report to the site administrators. Apart from using it as a standard web part, you can also use it in transformations to have the *Report abuse* link displayed with dynamically loaded content.

To see how the **In-line abuse report** web part is used, you can go to the **Examples -> Web parts -> Message boards -> Message board** page of the sample Corporate Site. As you can see in the screenshot below, the *Report abuse* link is included with each message on the board.



This is achieved using the In-line abuse report web part in the transformation used for displaying board messages. If you go to properties ( ) of the **Message board** web part and edit the **Message**

transformation (the *community.transformations.MessageBoard* transformation should be selected by default), you should see the same code as below.

```
<%@ Register Src="~/CMSModules/MessageBoards/Controls/MessageActions.ascx" TagName="
<%@ Register Src="~/CMSModules/AbuseReport/Controls/InlineAbuseReport.ascx" TagNam
<div class="CommentDetail">
    <asp:Panel ID="pnlRating" runat="server" CssClass="CommentRating" />
    <table width="100%">
        <tr>
            <td class="CommentUserName" style="width: 100%">
              <%# IfEmpty(Eval("MessageURL"), Eval("MessageUserName", true), "<a
href=\"" + Eval("MessageURL", true) + "\" target=\"_blank\">" + Eval
("MessageUserName", true) + "</a>")%>
            </td>
        </tr>
        <tr>
            <td class="CommentText">
              <%# CMS.GlobalHelper.TextHelper.EnsureLineEndings(Convert.ToString(Eva
            </td>
        </tr>
        <tr>
            <td class="CommentDate">
              <%# GetDateTime(Eval("MessageInserted")) %>
            </td>
        </tr>
        <tr>
            <td align="right" class="CommentAction">
              <cms:MessageActions ID="messageActions" runat="server" />
            </td>
        </tr>
        <tr>
            <td align="right" class="CommentAction">
                     <cms:AbuseReport ID="ucInlineAbuseReport" runat="server"
ReportObjectType="board.message" ReportObjectID='<%# Eval("MessageID") %>'
ReportTitle='<%# "Message board abuse report: " + Eval("MessageText") %>'
CMSPanel-SecurityAccess="AuthenticatedUsers" />
            </td>
        </tr>
    </table>
</div>
<hr style="border: 1px solid #CCCCCC;"/>
```

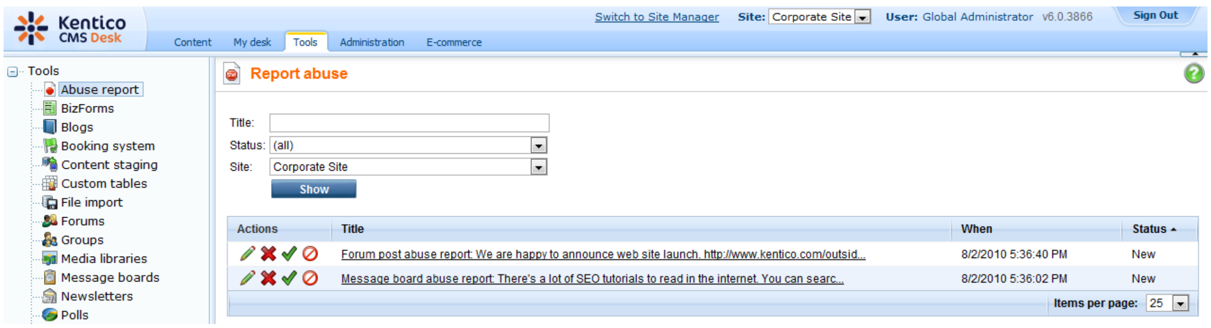
Let's take a closer look at the highlighted parts of the code. The highlighted line at the top is here to register the web part (actually control) so that it can be used in transformation code. As you can see, the web part code file is *~/CMSModules/AbuseReport/Controls/InlineAbuseReport.ascx*.

The second highlighted section is the actual web part, identified by the tag name and prefix defined in the *Register* tag. The *ReportObjectType="board.message"* and *ReportObjectID='<%# Eval ("MessageID") %>* parameters ensure that it will be possible to view details of a particular message (source of reported abuse) from the administration interface. These parameters pass the ID of the particular board message along with the report, so that the message can be identified and a link to it created. The *ReportTitle='<%# "Message board abuse report: " + Eval("MessageText") %>* parameter defines the title of the report displayed in the administration interface. The last parameter - *CMSPanel-SecurityAccess="AuthenticatedUsers"* - ensures that the link will be displayed only to authenticated users.

This way, you can use the web part in transformations of any document types you need. The only limitation is that the object details functionality is available only with **board messages**, **forum posts** and **blog comments**. Abuse reporting is still possible with other object types, but no object details are passed with the report and therefore can not be displayed in the administration interface.

8.3.4 Abuse reports management

Abuse reports can be viewed and managed in **CMS Desk -> Tools -> Abuse report**.



The top part of the page is a filter. By default, you get all reports displayed in the list below it. Using the filter, you can display only those reports that match the specified criteria. Available filtration parameters are **Title**, **Status** and **Site** from which the report was sent. To filter the reports, enter the appropriate parameters and click the **Show** button. A list of reports matching the criteria will be displayed.

The **Status** column shows which status is the report currently in. You can use the statuses to mark reports, so that **New**, **Solved** and **Rejected** reports can be distinguished from each other. By switching a report to a status, only the status changes - no other action happens. In other words, status switching is here just for your convenience, but you need to perform a suitable action manually.

The following actions are available for each of the listed reports:

- **Edit -** if clicked, the user will be redirected to report properties page, where you can edit the report's properties
- **Delete -** deletes the report
- **Mark as solved -** switches the report to the *Solved* status; used to mark reports for that the necessary actions have been taken
- **Mark as rejected -** switches the report to the *Rejected* status; used to mark reports that were not considered being cases of website abuse

If you click a title of a report, you will be redirected to the page that the report was sent from. If you mouse-over it, report description entered by the sender of the report will be displayed in form of a tooltip.

Editing a report

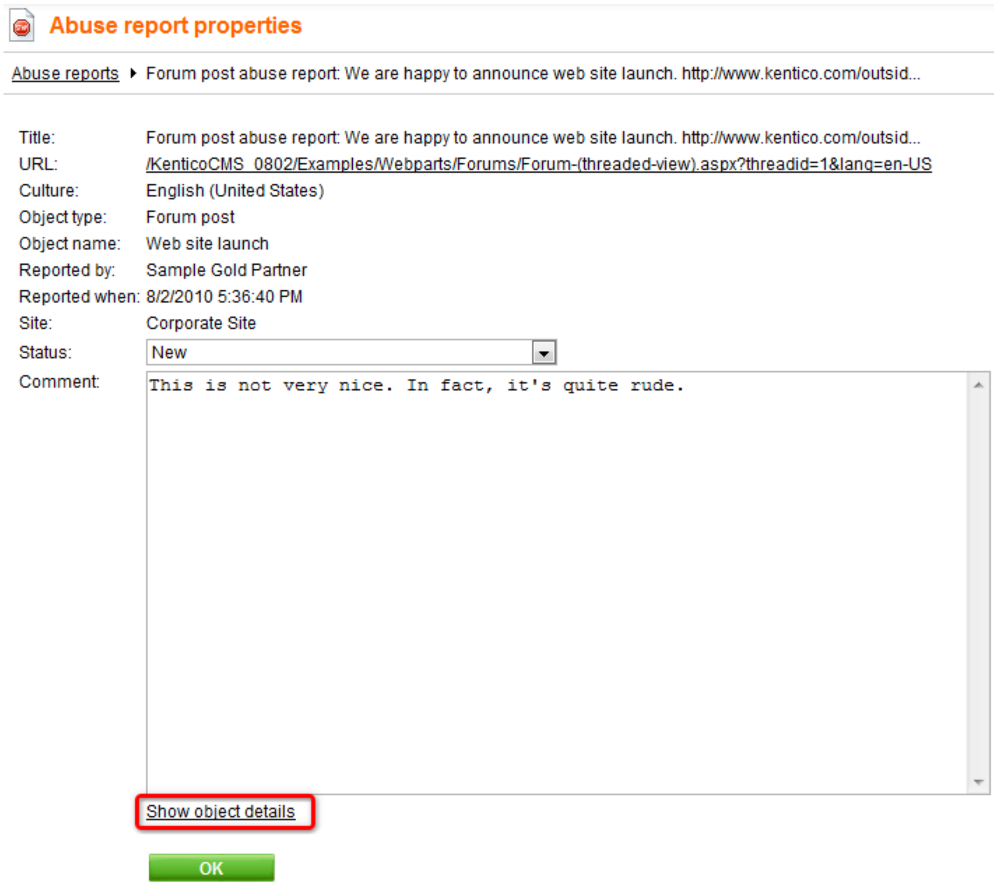
When editing ( ) a report, the following information is displayed:

Title	Title of the abuse report.

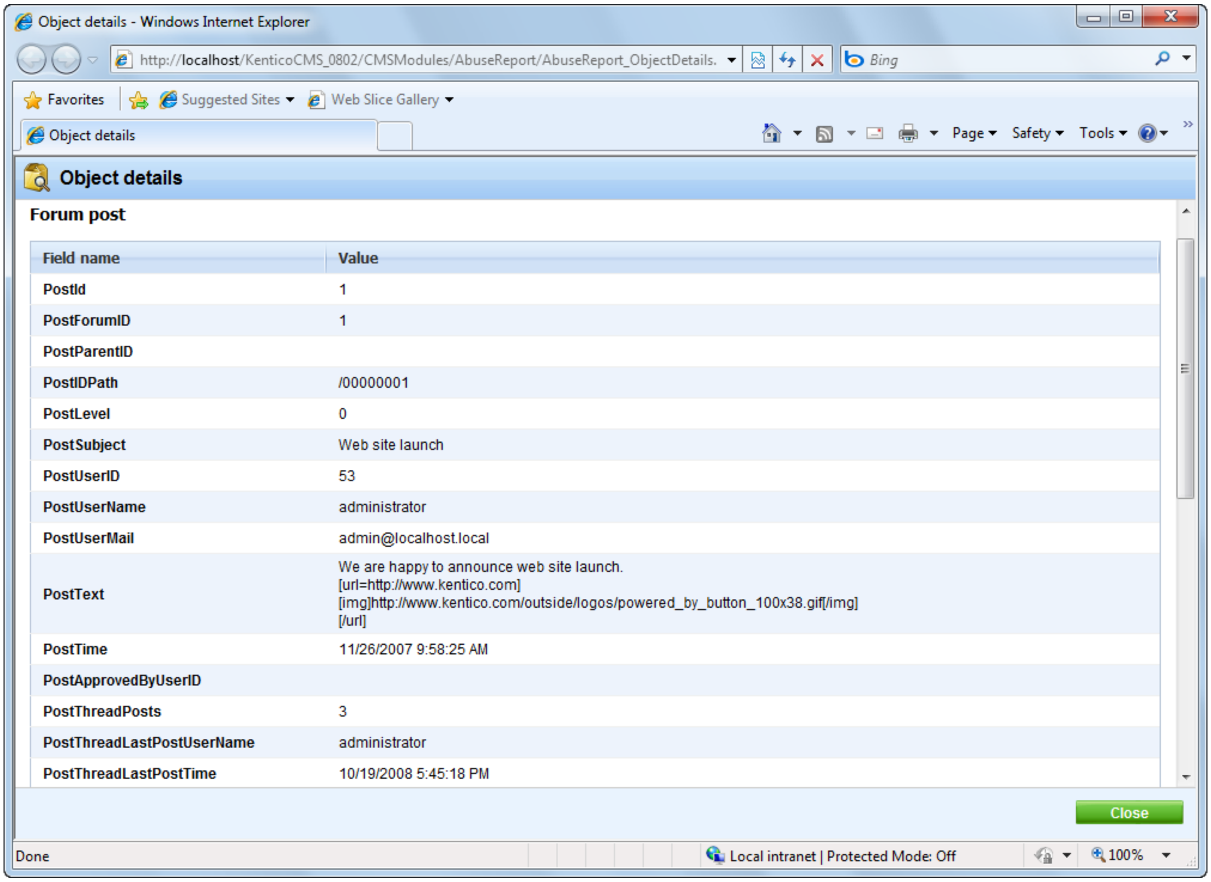
URL	URL of the page from which the report was sent. Click it to get redirected to that page.
Culture	Website culture from that the report was sent.
Object type	Type of object that was the cause of this report. If blank, the report was sent from some document.
Object name	Code name of the object that was the cause of this report.
Reported by	Site user who submitted the report.
Reported when	Time when the report was submitted.
Site	Website from that the report was submitted.
Status	Abuse report status, the following are possible: • **New -** the report is new and has not been solved yet • **Solved -** necessary actions have already been taken • **Rejected -** the report was not considered being a case of website abuse
Comment	Comment of the report entered by the reporting user.

Object details

If you edit () a report related to a **blog comment**, **board message** or **forum post**, the **Show object details** link is displayed below report comment, as highlighted in the screenshot above.



Clicking the link opens a new window with details about the source of the report (the board message, blog comment or forum post). This functionality is ensured automatically by the web parts listed in the [Integration with other modules](#) topic. You can also achieve it in transformations of these object types using the **In-line abuse report** web part, as described in the [Using the In-line abuse report web part in transformations](#) topic.



8.3.5 Integration with other modules

Apart from dedicated web parts described in the Available web parts chapter, the abuse report functionality is also embedded in the following web parts:

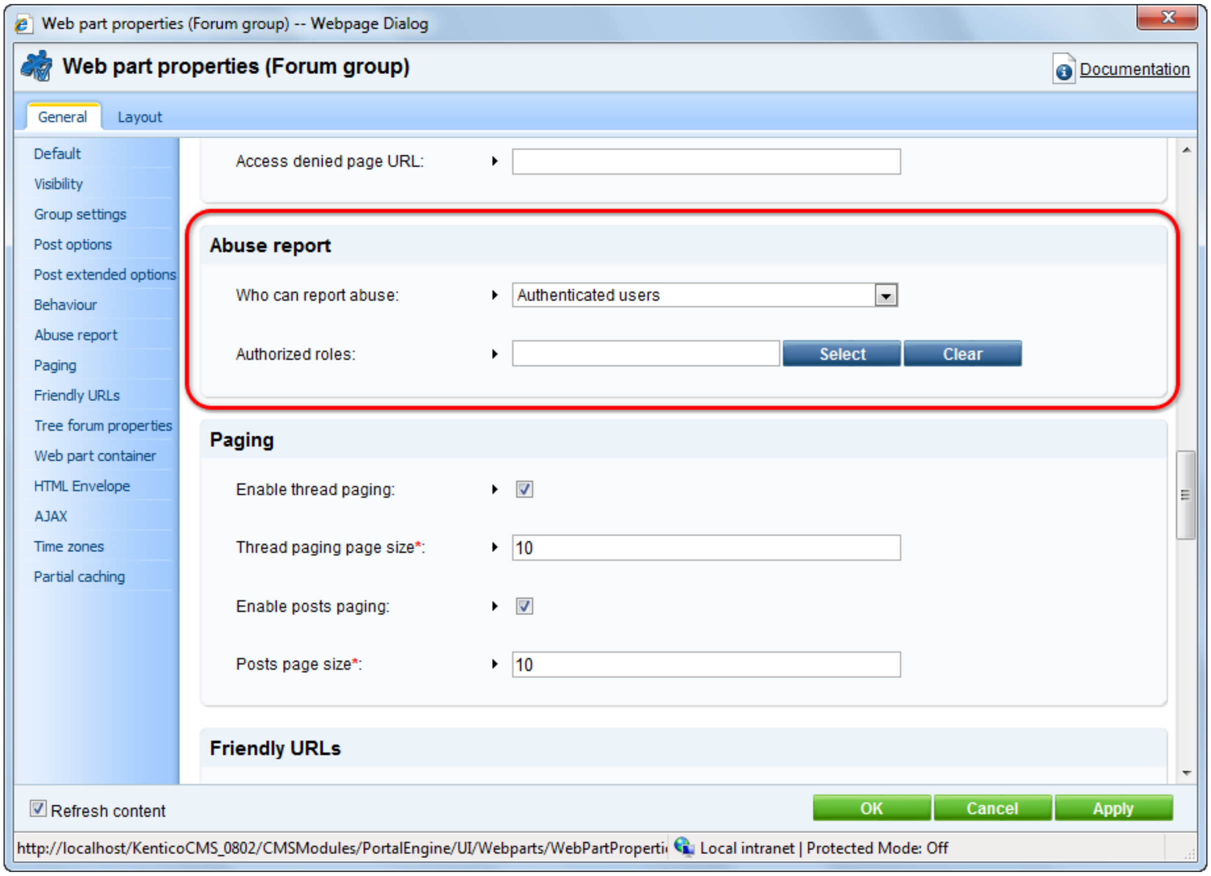
- **Blogs -> Comment view**
- **Community -> Group forum list**
- **Forums -> Forum group**
- **Forums -> Forum (Single forum - General)**
- **Forums -> Forum (Single forum - Tree layout)**
- **Forums -> Forum (Single forum - Flat layout)**

Each of these web parts has the **Abuse report** section in its properties, as depicted in the screenshot below. The section contains the following properties:

- **Who can report abuse** - using this property, you can determine to whom will the *Report abuse* link be displayed and who will therefore be able to submit abuse reports; the following options are available:
  - **All users** - the link will be displayed to anyone who views a page with the web part
  - **Authenticated users** - the link will be displayed only to authenticated users
  - **Authorized roles** - the link will be displayed only to users who are members of the roles specified by the *Authorized roles* property
  - **Nobody** - the link will not be displayed at all
- **Authorized roles** - if the *Who can report abuse* option is set to *Authorized roles*, you can use this

property to specify the roles to whose members the abuse report link will be displayed

For a complete reference on Kentico CMS web parts and their properties, please consult Kentico CMS Web Parts Reference.

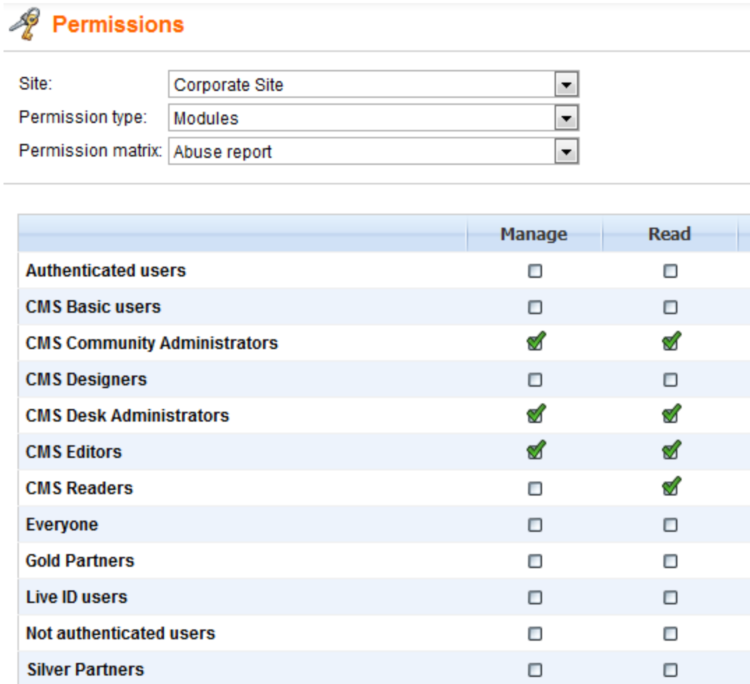


8.3.6 Security

Permissions for the Abuse report module can be set in **Site Manager -> Administration -> Permissions**. Select the **Modules -> Abuse report** permission matrix and grant appropriate permissions to particular roles.

The following permissions can be granted to the roles:

- **Manage** - members of the role are allowed to edit, delete, mark as solved and reject abuse reports
- **Read** - members of the role are allowed to view the abuse reports list



8.3.7 Abuse report internals and API

8.3.7.1 Database tables and API classes

The Abuse report module uses the following database table:

- **CMS_AbuseReport** - this is the database table used for storing submitted abuse reports

The Abuse report API is provided by the following **CMS.SiteProvider** namespace classes:

- **AbuseReportInfo**, **AbuseReportInfoProvider** - these classes provide functionality for managing abuse reports

The following topics show how methods from these classes can be used to manage abuse reports:

- Submitting an abuse report
- Getting abuse reports from the database
- Updating an abuse report
- Deleting an abuse report

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.3.7.2 Submitting an abuse report

The following code example shows how to create an abuse report and store it in the system database.

[C#]

```
using System;
using CMS.SiteProvider;
using CMS.CMSHelper;
using CMS.GlobalHelper;

...

        // Create a new abuse report
        AbuseReportInfo abuseReport = new AbuseReportInfo();

        abuseReport.ReportURL = UrlHelper.GetAbsoluteUrl(UrlHelper.CurrentURL);
        abuseReport.ReportCulture = CMSContext.PreferredCultureCode;

        abuseReport.ReportTitle = "Sample abuse report";
        abuseReport.ReportComment = "Sample comment";

        abuseReport.ReportWhen = DateTime.Now;
        abuseReport.ReportStatus = AbuseReportStatusEnum.New;
        abuseReport.ReportSiteID = CMSContext.CurrentSiteID;

        // Save the info object to the database
        AbuseReportInfoProvider.SetAbuseReportInfo(abuseReport);
```

8.3.7.3 Getting abuse reports from the database

The first code example on this page shows how you can get a single abuse report from the system database based on its ID.

[C#]

```
using CMS.SiteProvider;

...

        // Get abuse report as an info object by AbuseReportID
        AbuseReportInfo ari = AbuseReportInfoProvider.GetAbuseReportInfo(1);
```

The following code example shows how you can get a *DataSet* with multiple abuse reports. As you can see, two overrides of the *GetAbuseReports* method are used here. The first one gets the abuse reports based on the WHERE condition and ordered by the ORDER BY part of the SQL query. The second one does the same, while it also selects only the particular columns and retrieves only top N records (based on the order specified by the second parameter).

[C#]

```
using System.Data;
using CMS.SiteProvider;

...

        // Set the WHERE condition
        string where = "ReportStatus = 1";
        // Set the ORDER BY clause
        string orderBy = "ReportWhen DESC";
        // Set the colums to select
        string columns = "ReportID, ReportTitle, ReportURL, ReportComment, ReportSta
        // Set the number of rows to select
        int topN = 10;

        // Get a DataSet of abuse report info objects according to the given paramet
        DataSet ds = AbuseReportInfoProvider.GetAbuseReports(where, orderBy);

        DataSet ds2 = AbuseReportInfoProvider.GetAbuseReports(where, orderBy, column
```

8.3.7.4 Updating an abuse report

The following code example shows how you can modify details of an existing abuse report.

[C#]

```
using CMS.SiteProvider;

...

        // First get the abuse report info by its ID
        AbuseReportInfo ari = AbuseReportInfoProvider.GetAbuseReportInfo(1);

        if (ari != null)
        {
            // Set abuse report status to solved
            ari.ReportStatus = AbuseReportStatusEnum.Solved;
            // Edit comment
            ari.ReportComment += " - SOLVED";
            // Save the abuse report info to the database
            AbuseReportInfoProvider.SetAbuseReportInfo(ari);
        }
```

8.3.7.5 Deleting an abuse report

The following code example shows how you can delete an abuse report from the system database.

[C#]

```
using CMS.SiteProvider;
```

```
...

        // First get the abuse report info by its ID
        AbuseReportInfo ari = AbuseReportInfoProvider.GetAbuseReportInfo(1);

        if (ari != null)
        {
            // Delete the abuse report
            AbuseReportInfoProvider.DeleteAbuseReportInfo(ari);
        }
```

8.4 Alternative forms

8.4.1 Overview

The Alternative forms module enables layout and functionality customization of existing **BizForms**, **document types**, **system tables** and **custom tables**. These alternatives can then be easily used instead of the originals when needed.

8.4.2 Creating an alternative form

This example shows how to create an alternative form of the existing **Contact us** BizForm on the sample Corporate Site.

Alternative forms of document types, system tables and custom tables can be created in exactly the same way as described in this example. You only need to access the alternative forms tab from the following locations:

- **Site manager -> Development -> Custom tables**
- **Site manager -> Development -> Document types**
- **Site manager -> Development -> System tables**

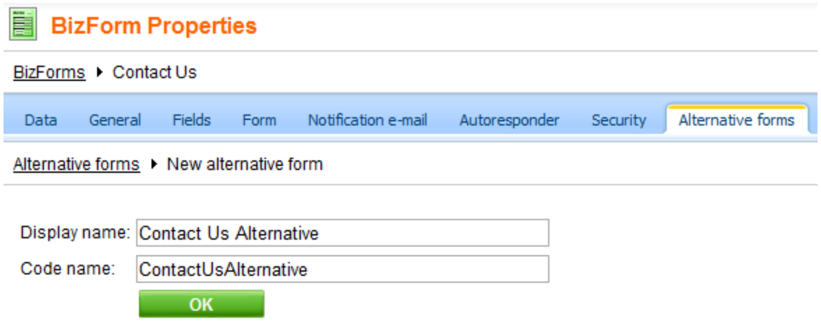
There, just choose to **Edit** ( ) the particular item, switch to its **Alternative forms** tab and follow the instructions below, starting from the second step.

1. Go to **CMS Desk -> Tools -> BizForms**. Choose to **Edit** ( ) the **Contact us** BizForm. The **BizForm Properties** screen will appear.

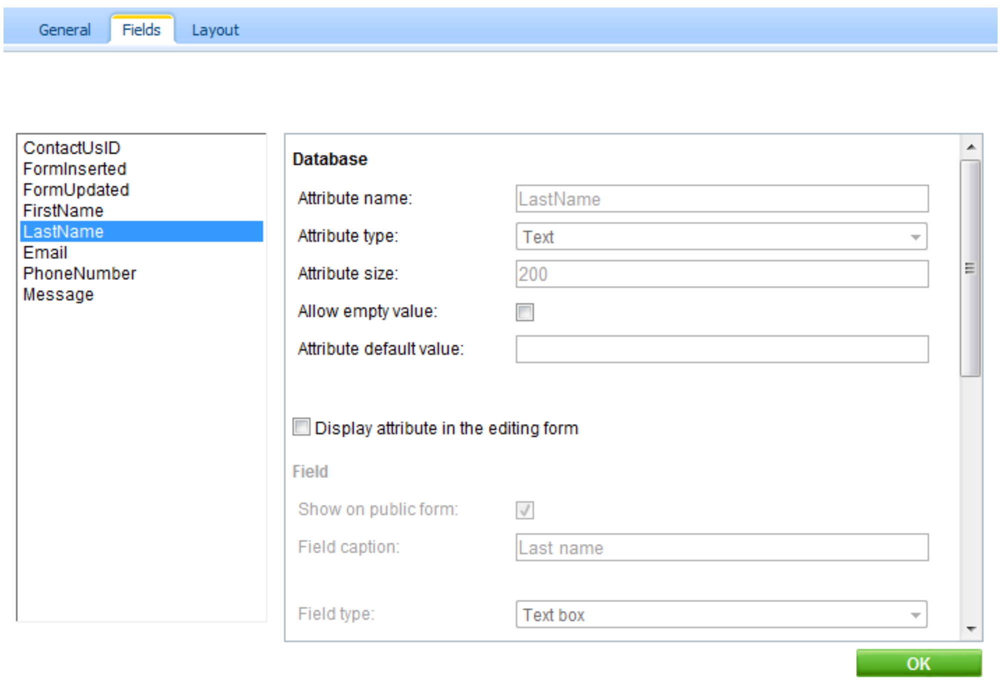
2. On **BizForm Properties**, switch to the **Alternative forms** tab and choose to **Create new form**.

3. On the form that will be displayed, enter the following details and click **OK**.

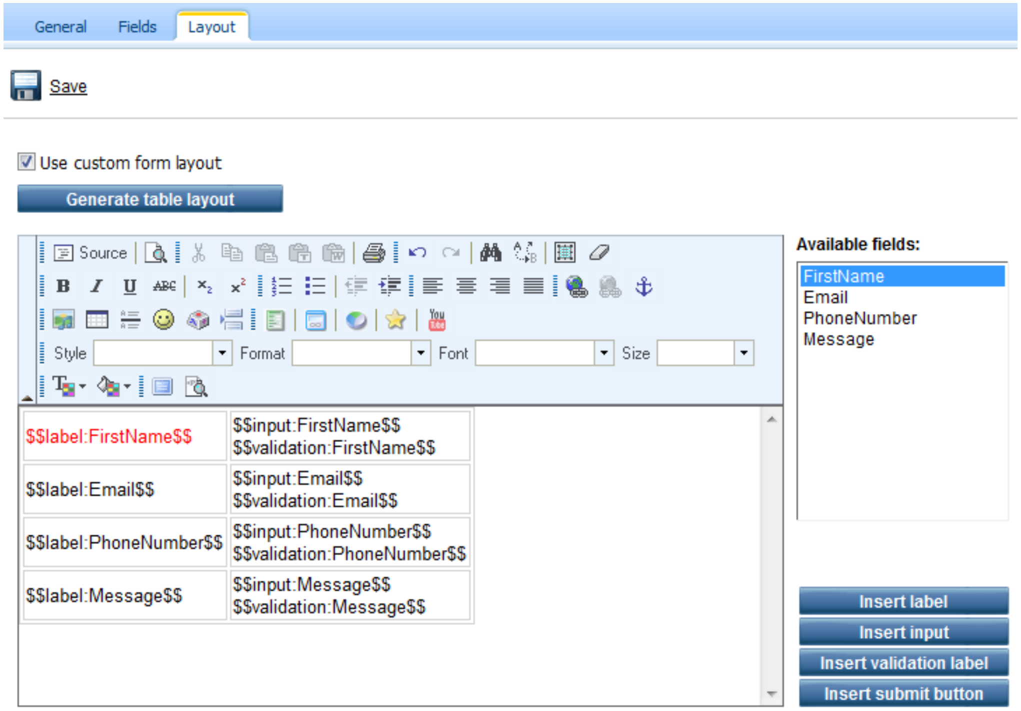
- **Display name:** Contact Us Alternative
- **Code name:** ContactUsAlternative



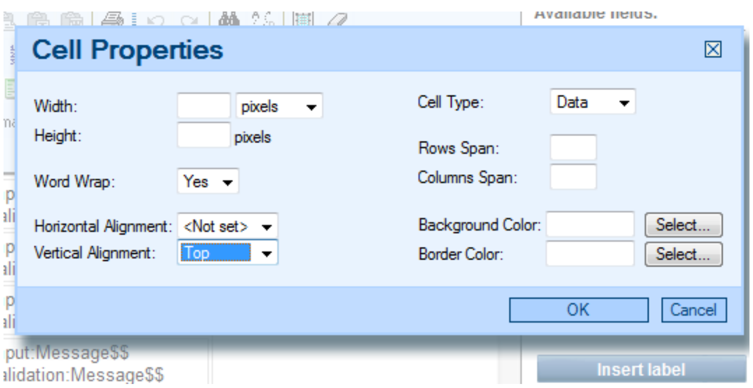
4. Switch to the **Fields** tab. As you can see, all fields present in the original form are present here as well and you can now make modifications to them. Let's disable the **LastName** field as an example. Select the field from the list on the left and on it's properties, uncheck the **Display attribute in the editing form** check-box. Confirm by clicking **OK**. Like this, you can modify any field in the form according to your needs.



5. Not only the fields, but also the layout of the form can be modified. Switch to the **Layout** tab and check the **Use custom layout** check-box. The layout editor will appear. Notice that the **LastName** field that we disabled in the previous step is not offered in the **Available fields** listbox. Click the **Generate table layout** button. Table layout will be generated in the editing field below. Highlight the $$label: FirstName$$ text in the first row and change its color to red.

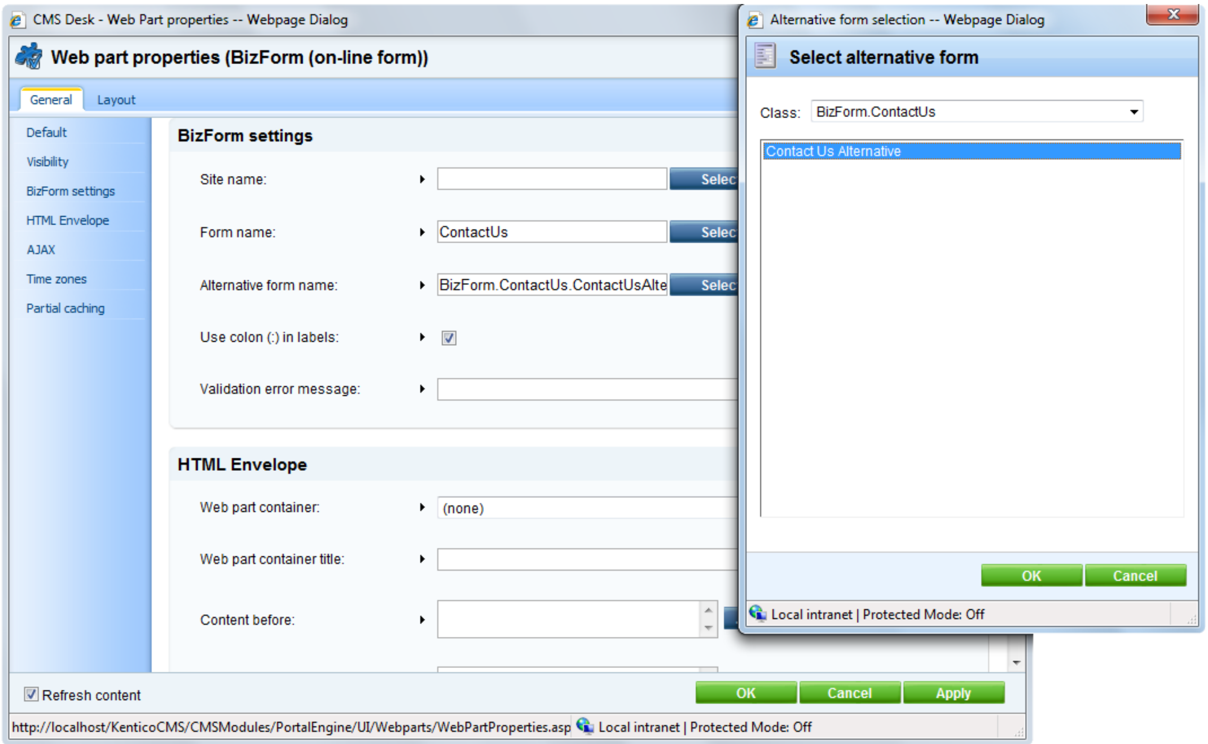


Also select the rows of the first table column, right click it and from the context menu, choose **Cell -> Cell properties**. In the displayed dialog, choose **Vertical alignment: Top** and click **OK**. This will make the labels to be aligned to their fields. Click **Save**.

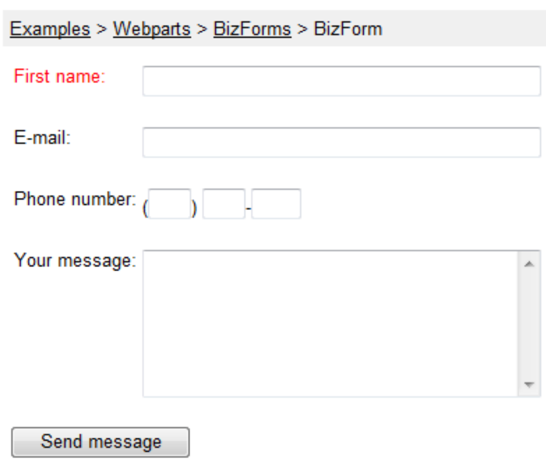


6. Let's take a look at what we've created. Switch to **CMS Desk -> Content**. From the content tree, select **Examples -> BizForms -> BizForm**. As you can see, there is the original version of the **Contact Us** form present on the page. We will edit the web part's properties so that the alternative from is used instead.

7. In **Edit** mode, switch to the page's **Design** tab and choose to configure the web part properties. We will be concerned about the **Alternative form name** field. Click the **Select** button next to it and from the list, select our **Contact Us Alternative**. Click **OK**.



8. Now when you switch to the **Live site** mode, you should see the modified version as in the following screenshot:

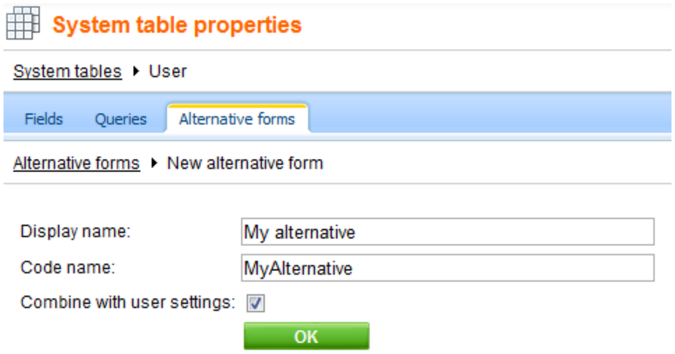


You have learned how to create an alternative form to an existing BizForm and use it on your website.

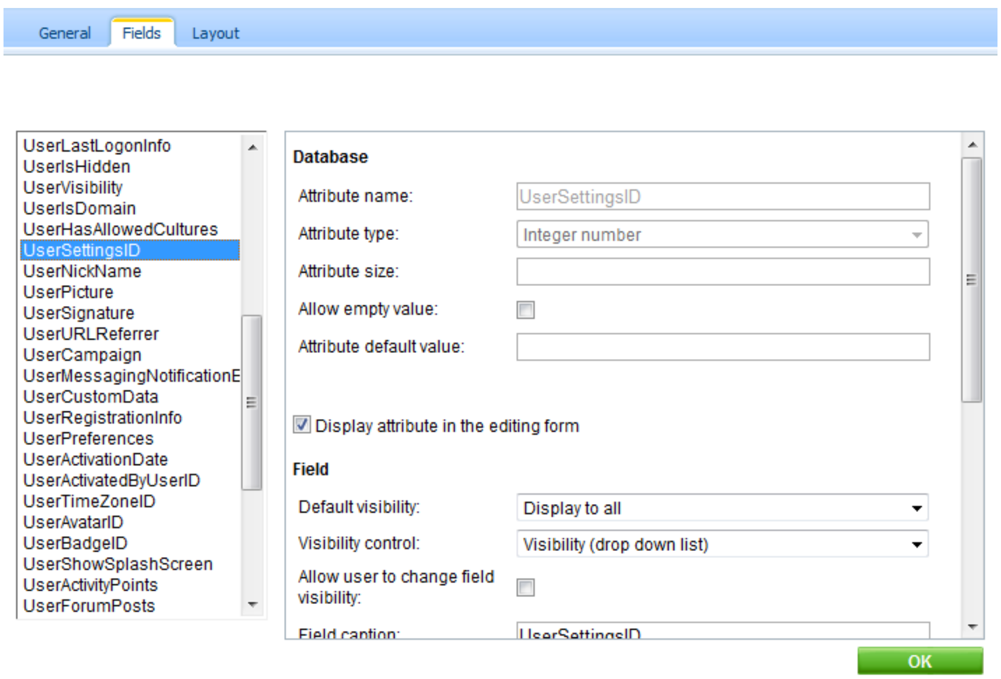
8.4.3 Joining two classes into one form

It is also possible to join two classes into one alternative form. This option is currently available only for the **User** and **User - Settings** system tables.

1. When creating an alternative form of the **User** system table, you have the option to check the **Combine with user settings** check-box. This causes that fields contained in the **User - Settings** table will be included in the alternative form.



2. Now if you switch to the **Fields** tab, you will see that besides the original fields contained in the **Users** table, fields from the **User - Settings** table are also present, as you can see in the following screenshot:



8.4.4 Automatically used alternative forms

If you create alternative form and give it one of the following code names, the form will be automatically used when performing the corresponding action. The following table shows the names and actions when the forms are used.

Alternative form Code name	Used when	Supported by
insert	creating a new document	document types, BizForms, system tables, custom tables
update	editing an existing document	document types, BizForms, system tables, custom tables

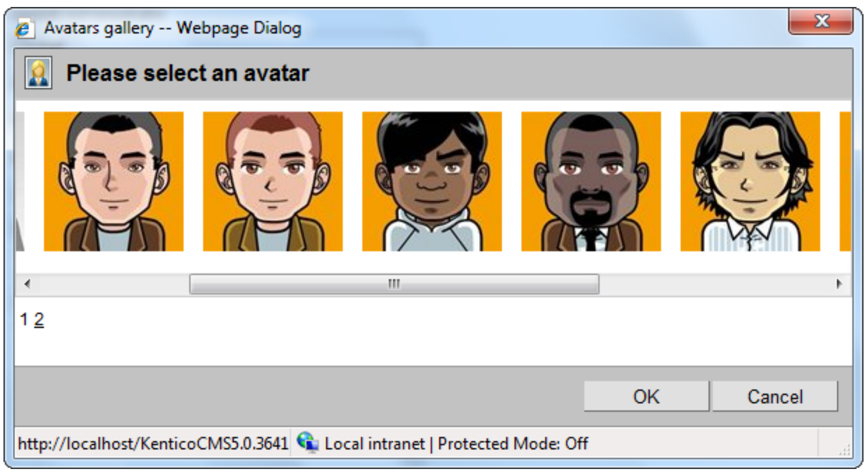
newculture	creating a new culture version of a document	document types

8.5 Avatars

8.5.1 Overview

The Avatars module enables users to have an image associated with their account. This image is called an **avatar** and is displayed on the user's public profile, in forum posts, etc. An avatar serves as a graphical representation of a user, and is used to personalize that user's contributions on websites. More information about users in general can be found in the Development -> Membership, permissions and security chapter of this guide.

Users can either choose an avatar from a gallery of pre-defined avatars if this option is enabled or create a custom avatar by uploading their own image from a file on their local disk. All common image formats are supported, including animations.



Unlike pre-defined avatars, custom avatars can't be selected by other users and they are deleted from the system if the user they were uploaded by changes her avatar.

Groups can also have avatars, these are displayed on the group's profile and can benefit the group by providing a way for it to be identified better and faster, etc. To read more about groups, please refer to the Modules -> Groups chapter of this guide.

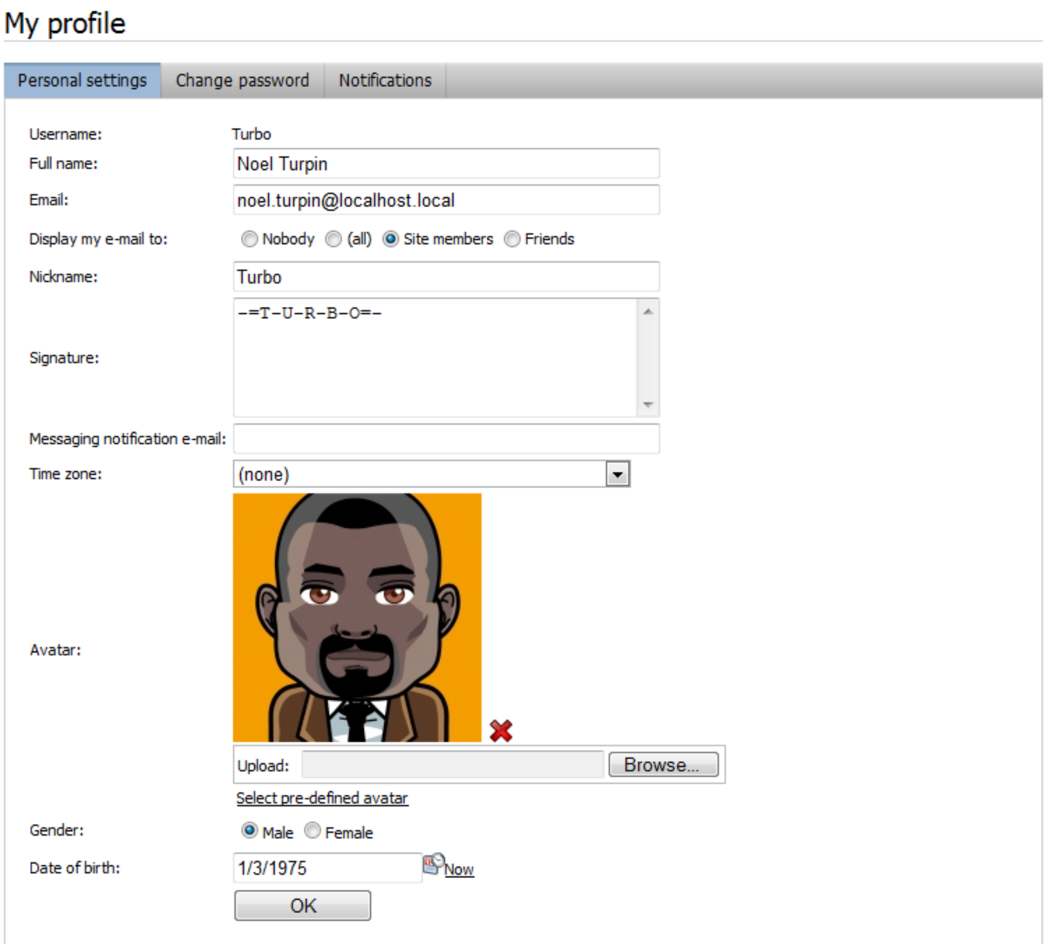
There are several ways for users or group administrators to add or change avatars. See the Changing user avatars or Changing group avatars topics for more details.

Site administrators can manage all avatars or change their settings. This is described in the Managing avatars and Settings topics.

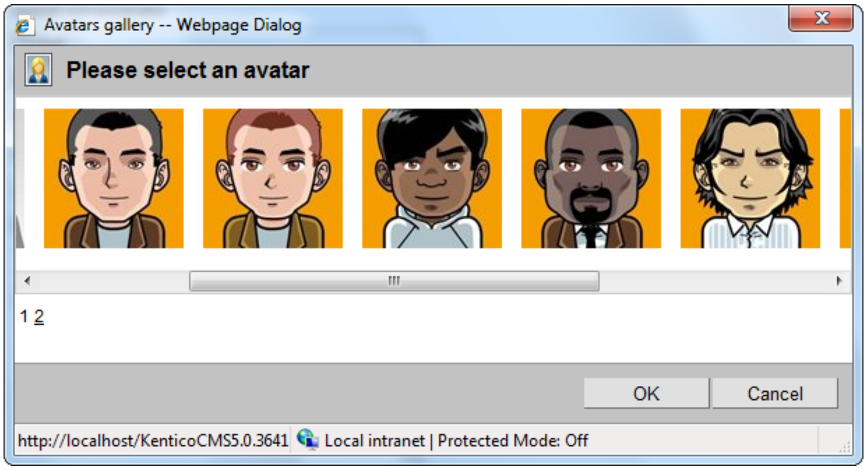
The Avatars internals and API sub-chapter provides information about the database tables and classes used by the module, examples of how avatars can be managed using the API and how they can be displayed in transformations.

8.5.2 Changing user avatars

When a new user registers on a site, the default avatar will be assigned to them. After that, they can change thier avatar using the **My account** web part. It can be done on its **Personal settings** tab, as you can see in the screenshot below.



Users can **Delete** (✖) the avatar or **Upload** a custom one from a file. If pre-defined avatars are enabled in site settings, users can also click the **Select pre-defined avatar** link, which displays a gallery of pre-defined avatars.

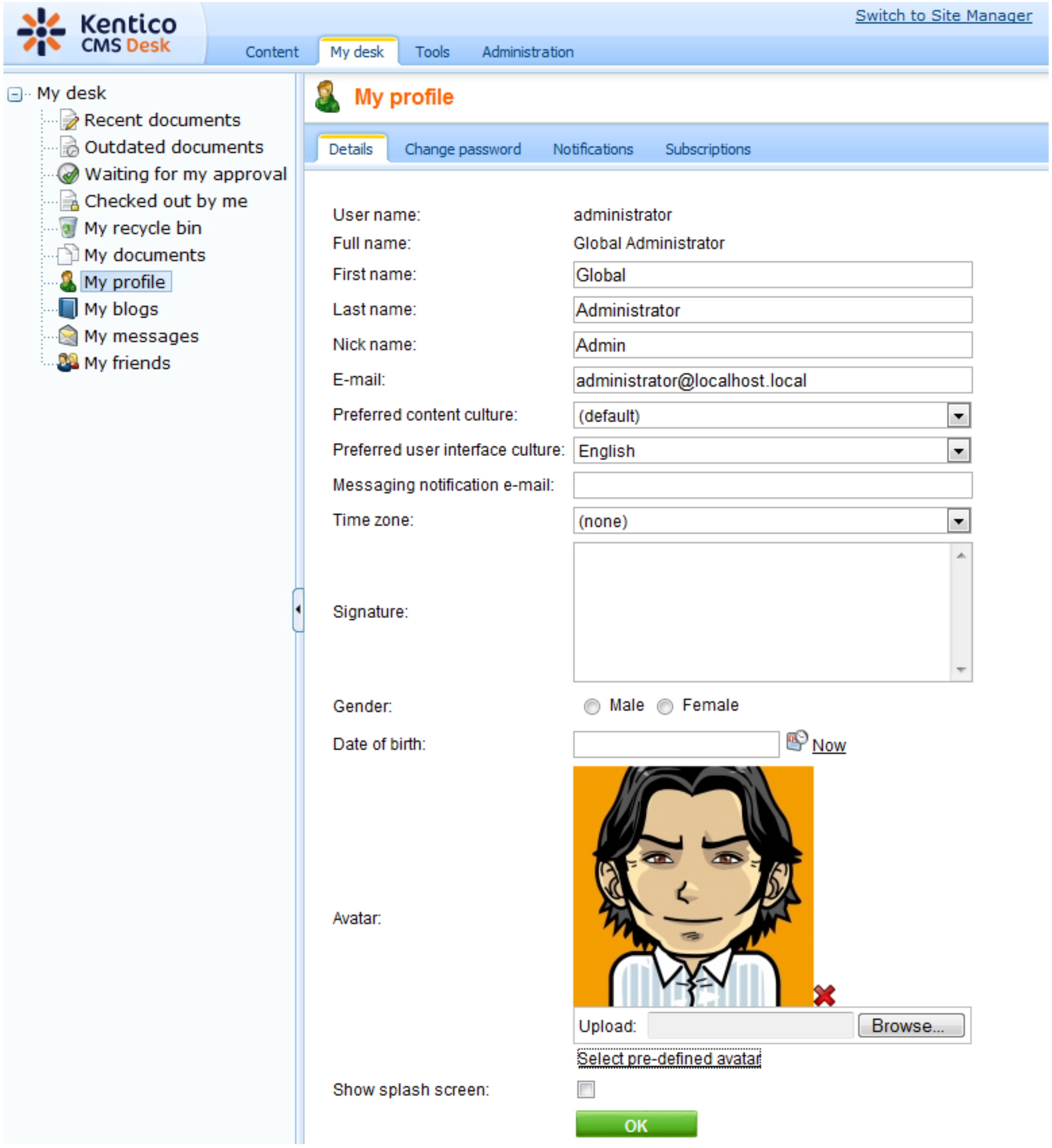


You can find a live example of this on our **Community starter site**. Just log-in as some of the pre-defined users (e.g. *Turbo* with blank password) and click the **Edit my profile** link in the **Shortcuts** menu on the right.

Changing user avatar in CMS Desk

Users with access to **CMS Desk** can change their avatars in **My Desk -> My profile**. It can be done the same way as described above. You can **Delete** (✖) the avatar or **Upload** your own avatar from a file.

If default avatars are enabled in site settings, you can also click the **Select pre-defined avatar** link, which displays a gallery of pre-defined avatars from which you can easily pick one by clicking it and clicking **OK**.

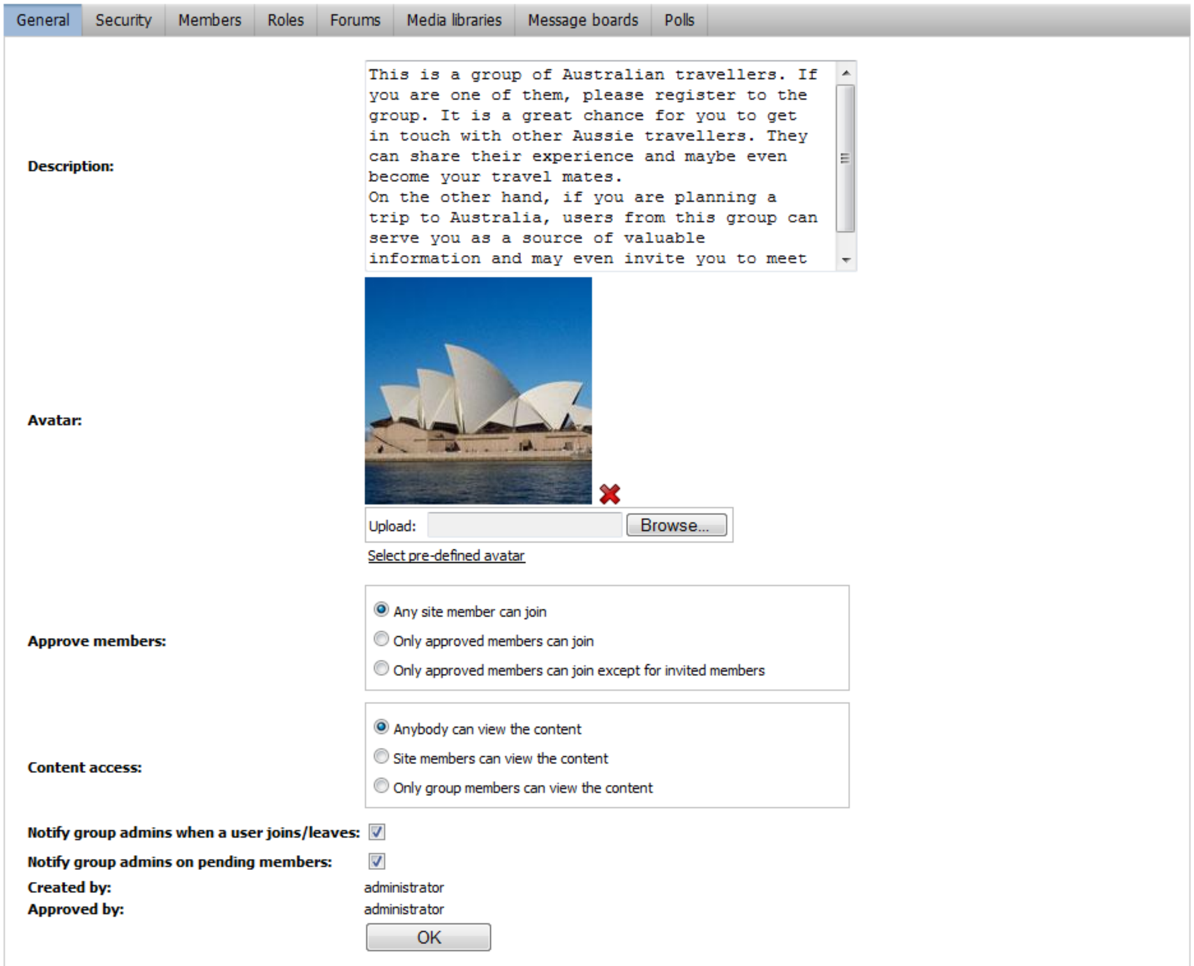


8.5.3 Changing group avatars

When a new group is created, the default avatar will be assigned to it. After that, group administrators can change the group's avatar using the **Group profile** web part.

Users can **Delete** (✖) the avatar or **Upload** a custom one from a file. If pre-defined avatars are enabled in site settings, users can also click the **Select pre-defined avatar** link, which displays a gallery of pre-defined avatars.

You can find a live example of this on the **Community starter site**. Sign-in as some group administrator (e.g. *Josh* with blank password, he is the *Australian travellers* group admin) and click Groups in the main menu. You should see the **Australian travellers** group in the **My groups** section. Click it, and what you can see in the screenshot below will be displayed to you.

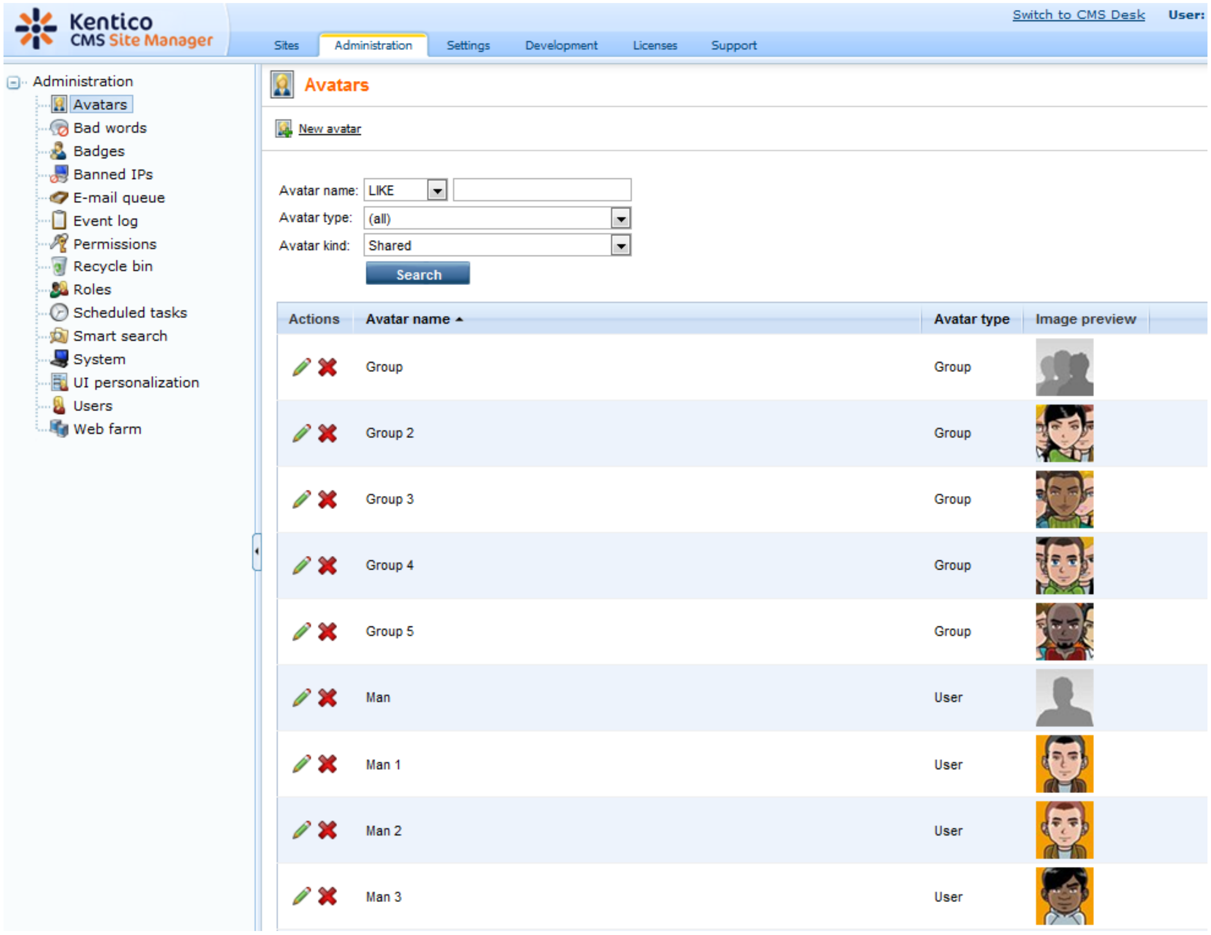


8.5.4 Managing avatars

The administration interface for avatar management is located in **Site Manager -> Administration -> Avatars**.

You can filter displayed avatars using the filter above the list. Possible filtering parameters are **Avatar name**, **Avatar type** (user or group avatar, avatars of type *all* can be used for both) and **Avatar kind** (shared avatars are the pre-defined ones, while custom avatars are those that users uploaded from a file). Click **Search** to display only avatars matching the selected criteria.

You can **Edit** ( ) or **Delete** ( ) listed avatars.



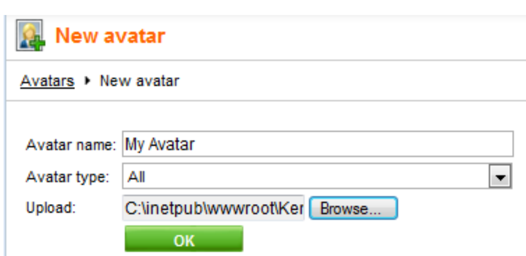
Creating pre-defined avatars

1. New **pre-defined avatars** can be created using the **New avatar** link at the top part of the page. Click it.

2. You will be asked to enter the following details:

- **Avatar name** - name of the avatar
- **Avatar type** - choose if the avatar can be used for users, groups or both
- **Upload** - enter the path to the avatar image on your local machine or click the **Browse** button to browse and locate the file

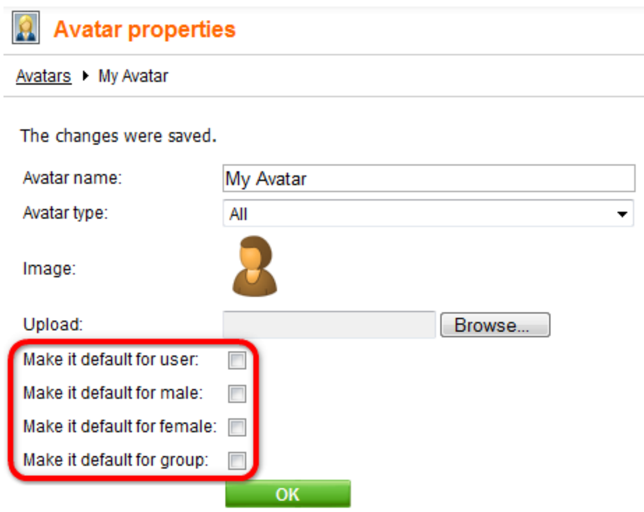
When entered, click **OK** to proceed.



3. The avatar is now created and if you go back to the list of avatars, you should see the avatar present in the list. However, you can set the following extra properties of the avatar now or any time later when editing the avatar:

- **Make it default for user** - if checked, this avatar will be the default avatar for users
- **Make it default for male** - if checked, this avatar will be the default avatar for male users
- **Make it default for female** - if checked, this avatar will be the default avatar for female users
- **Make it default for group** - if checked, this avatar will be the default avatar for groups

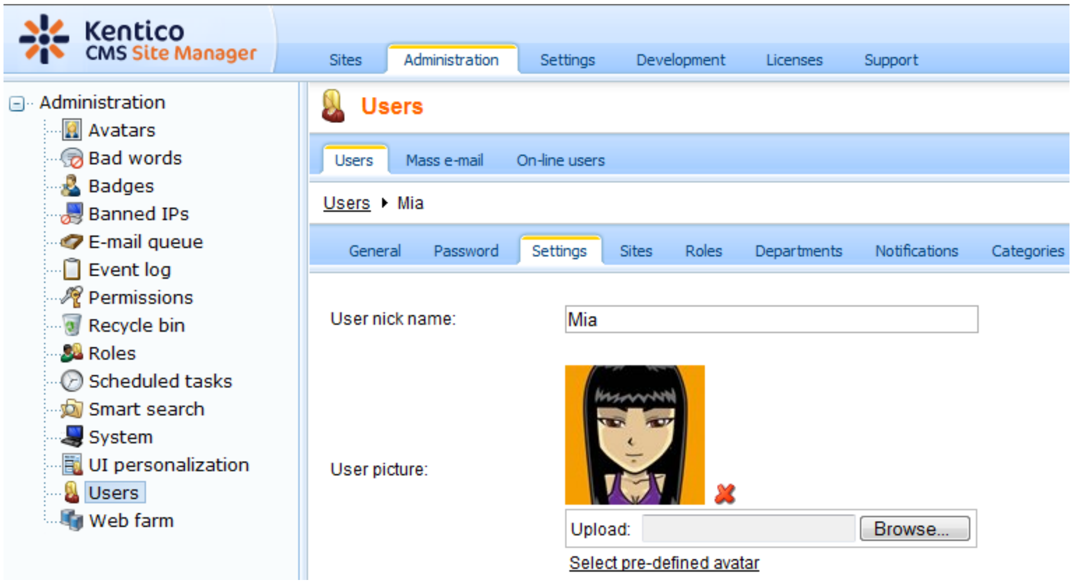
Default avatars will be assigned to newly created users or groups automatically when the user or group is created.



Administrating users' avatars

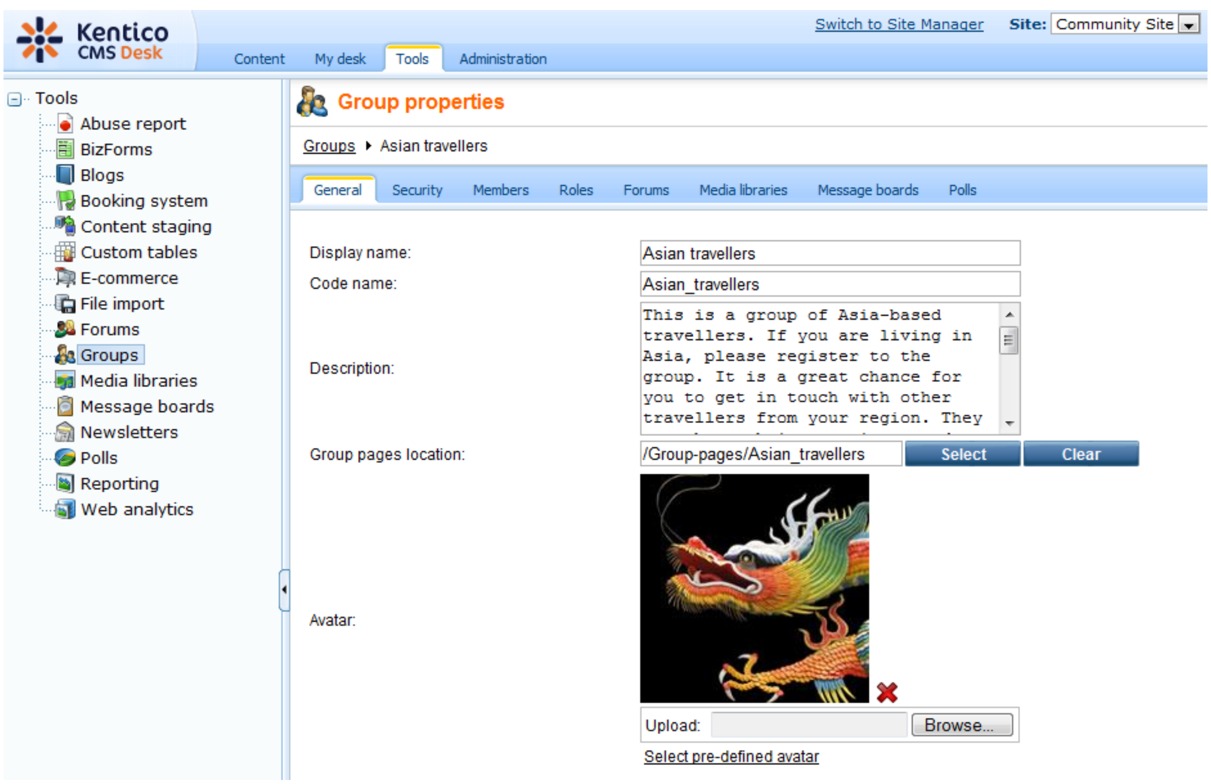
Site administrators can change the avatar of any user. If you go to **Site Manager -> Administration -> Users**, choose to **Edit** () some of the users in the list and switch to their **Settings** tab, you should see the user's avatar in the **User picture** field, as you can see in the screenshot below. You can **Delete** () the avatar or **Upload** your own avatar from a file.

If default avatars are enabled in site settings, you can also click the **Select pre-defined avatar** link, which displays a gallery of pre-defined avatars from which you can easily pick one by clicking it and clicking **OK**.



Changing group avatars in CMS Desk

Site administrators can change the avatar of any group. If you go to **CMS Desk -> Tools -> Groups** and choose to **Edit** (✎) some of the groups, you should see the group's avatar in the **Avatar** section, as depicted in the screenshot below. A Group avatar can be **Deleted** (✖), a new one can be **Uploaded** from a file or selected from a gallery of pre-defined avatars (if this is enabled in site settings).



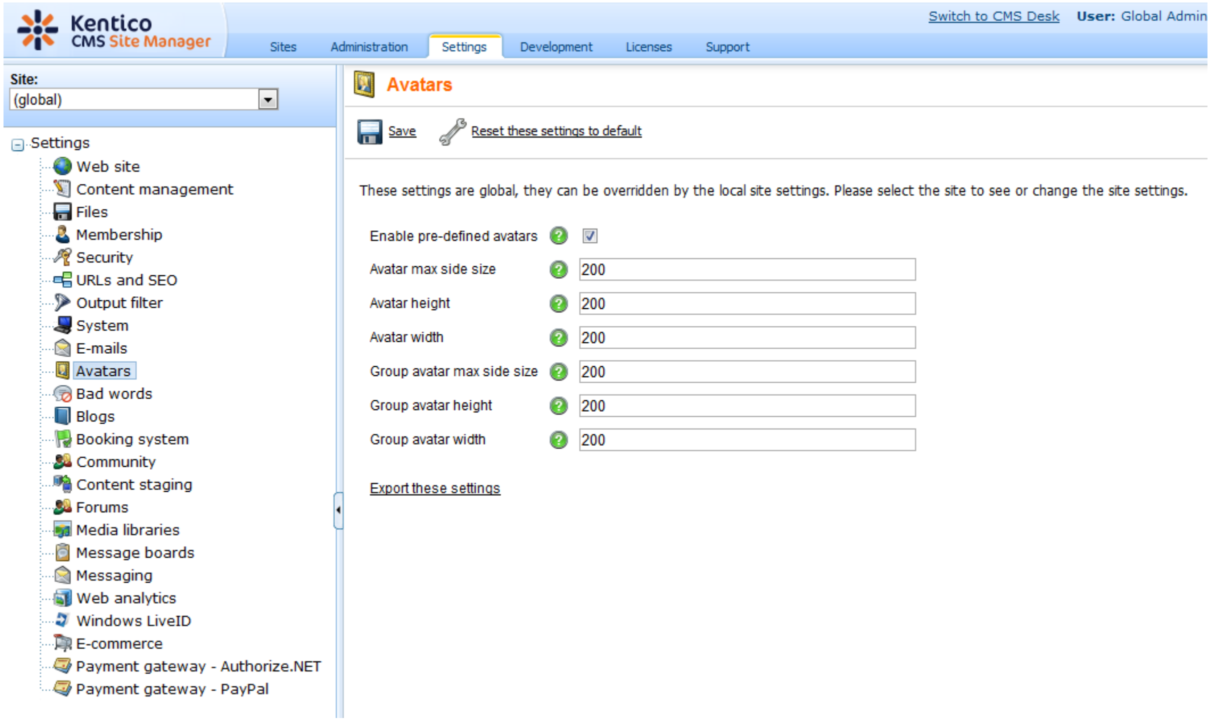
8.5.5 Settings

Settings of the Avatars module can be done in **Site Manager -> Settings -> Avatars**. The following settings are available:

- **Enable pre-defined avatars** - if checked, default avatars can be selected when selecting a user's or group's avatar; if unchecked, only custom uploaded avatars can be used

- **Avatar max side size** - maximal size of **user avatars**; if one or both sides of the image are longer, the image will be resized so that the longer side's size matches the entered value; if 0 is entered, *Avatar height* and *Avatar width* values will be used instead
- **Avatar height** - if *Avatar max side size* is set to 0, images will be resized to this height
- **Avatar width** - if *Avatar max side size* is set to 0, images will be resized to this width

- **Group avatar max side size** - maximal size of **group avatars**; if one or both sides of the image are longer, the image will be resized so that the longer side's size matches the entered value; if 0 is entered, *Group avatar* height and *Group avatar width* values will be used instead
- **Group avatar height** - if *Group avatar max side size* is set to 0, images will be resized to this height
- **Group avatar width** - if *Group avatar max side size* is set to 0, images will be resized to this width



8.5.6 Avatars internals and API

8.5.6.1 Database tables and API classes

The Avatars module uses the following database table:

- **CMS_Avatar** - this is the database table used for storing avatars

The Avatars API is provided by the following **CMS.SiteProvider** namespace classes:

- **AvatarInfo**, **AvatarInfoProvider** - these classes provide functionality for managing avatars

The following topics show examples of how these classes can be used:

- Getting avatar data
- Modifying and creating avatars

The following topic contains examples of how avatars can be displayed in transformations:

- Displaying avatars in transformations

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.5.6.2 Getting avatar data

The following sample code shows how you can get an avatar as an *AvatarInfo* object:

[C#]

```
using CMS.SiteProvider;

...

        // Get avatar object by AvatarID
        AvatarInfo ai = AvatarInfoProvider.GetAvatarInfo(1);


        // Get avatar object by AvatarName
        AvatarInfo ai2 = AvatarInfoProvider.GetAvatarInfo("AvatarName");
```

The following sample code shows how you can get a DataSet containing all avatars from the system with an *ID* greater than 5 ordered by their name:

[C#]

```
using System.Data;
using CMS.SiteProvider;

...

        string where = "AvatarID > 5";
        string orderBy = "AvatarName";

        // Get DataSet of avatar objects according to the given parameters
```

```
        DataSet ds = AvatarInfoProvider.GetAvatars(where, orderBy);
```

8.5.6.3 Modifying and creating avatars

The following sample code shows how you can modify the name of an existing avatar:

[C#]

```
using CMS.SiteProvider;

...

        // Get avatar object by AvatarID
        AvatarInfo ai = AvatarInfoProvider.GetAvatarInfoWithoutBinary(avatarID);

        // Change avatar name
        ai.AvatarName = ai.AvatarName + "_Updated";

        // Save avatar object
        AvatarInfoProvider.SetAvatarInfo(ai);
```

The following sample code shows how you can create a new pre-defined avatar of type 'All' and add it to the system:

[C#]

```
using CMS.SiteProvider;

...

        // Create new avatar object
        AvatarInfo ai = new AvatarInfo("\\Directory\\custom.jpg");

        // Set properties
        ai.AvatarName = "New avatar";
        ai.AvatarType = AvatarInfoProvider.GetAvatarTypeString(AvatarTypeEnum.All);
        ai.AvatarIsCustom = false;

        // Save object to database
        AvatarInfoProvider.SetAvatarInfo(ai);
```

8.5.6.4 Displaying avatars in transformations

User avatars

User avatars can also be displayed in transformations by using the *<%# GetUserAvatarImage(...) %>* method. Here are some examples of how it can be used:

The following method returns the image tag of the avatar contained in the *AvatarGuid* field of the currently transformed user, with a maximum *sidesize* of 50 px. and *Alt tag* equal to the user's *Nickname* or *Username.* The *AvatarGuid* field must be accessible for this method to work:

```
<%# GetUserAvatarImage(50, HTMLEncode(GetNotEmpty("UserNickname;UserName"))) %>
```

The following method returns the image tag of the avatar with ID *UserAvatarID*, of user with ID *UserID*, with a maximum *sidesize* of 50 px. and *Alt tag* equal to the user's *Username*:

```
<%# GetUserAvatarImage(Eval("UserAvatarID"), Eval("UserID"), 50, Eval("UserName"))
%>
```

The following method returns the image tag of the avatar with ID *UserAvatarID*, of user with ID *UserID*, with a maximum *width* of 40 px., maximum *height* of 45 px. and *Alt tag* equal to the user's *Username*:

```
<%# GetUserAvatarImage(Eval("UserAvatarID"), Eval("UserID"), 0, 40, 45, Eval(
"UserName")) %>
```

Group avatars

Group avatars can by displayed using the *<%# GetGroupAvatarImage(...) %>* method. Here are some examples:

The following method returns the image tag of the group avatar contained in the *AvatarGuid* field of the currently transformed group, with a maximum *sidesize* of 50 px. and *Alt tag* equal to the group's *Display name.* The *AvatarGuid* field must be accessible for this method to work:

```
<%# GetGroupAvatarImage(50, Eval("GroupDisplayName", true)) %>
```

The following method returns the image tag of the group avatar with ID *GroupAvatarID*, with a maximum *sidesize* of 50 px. and *Alt tag* equal to the group's *Display name*:

```
<%# GetGroupAvatarImage(Eval("GroupAvatarID"), 50, Eval("GroupDisplayName", true))
%>
```

The following method returns the image tag of the group avatar with ID *GroupAvatarID*, with a maximum *width* of 40 px., maximum *height* of 45 px. and *Alt tag* equal to the group's *Display name*:

```
<%# GetGroupAvatarImage(Eval("GroupAvatarID"), 0, 40, 45, Eval("GroupDisplayName",
true)) %>
```

Image tag

The following is an example of an image tag generated by the *<%# GetUserAvatarImage(...) %>*

method. As you can see, the method uses the **~/CMSModules/Avatars/CMSPages/GetAvatar.aspx** page to get the source image.

```
<img style="border-width: 0px;" alt="Andy" src="/CMSModules/Avatars/CMSPages/
GetAvatar.aspx?avatarguid=42f9d97e-b0c8-43a2-8a35-62dfec5ae64a&amp;maxsidesize=50"
/>
```

8.6 Bad words

8.6.1 Overview

The Bad words module can be used as a filter for unwanted input from website users. This can be anything from rude language to spam or links to illegal content, basically anything that is detectable by presence of some keyword in input text.

The Bad words module is capable of filtering input submitted via the following modules:

- Forums - filters forum posts
- Blogs - filters blog comments
- Messaging - filters messages sent via the module
- Message boards - filters messages posted on particular boards

For the module to work, you first need to enable it as described in Enabling the module. Filtering is then performed based on keywords - so called bad words - defined by site administrators in **Site manager -> Administration -> Bad words**. The procedure of defining bad words is described in the Defining a bad word topic.

Actions	Expression ▲	Action	Replacement	All cultures
	arse	Remove		Yes
	asshole	Replace (default)	***** (default)	Yes
	assramer	Request moderation		Yes
	bastard	Deny		Yes

If a user inputs some text containing one or more of these words, a predefined action is performed. Available actions range from removing the word, replacing it with a pre-defined string or just reporting abuse to website administrators by means of the Abuse report module. All actions and their effects are described in the Possible actions topic.

By default, input from all users except global administrators is filtered. However, you may grant the **Use bad words** permission to some roles, which will disable filtering of input submitted by its members. More information on this option can be found in the Security topic.

The Bad words internals and API sub-chapter provides information about the database tables and classes used by the module, as well as examples of how bad words can be managed and bad word checks performed using Kentico CMS API.

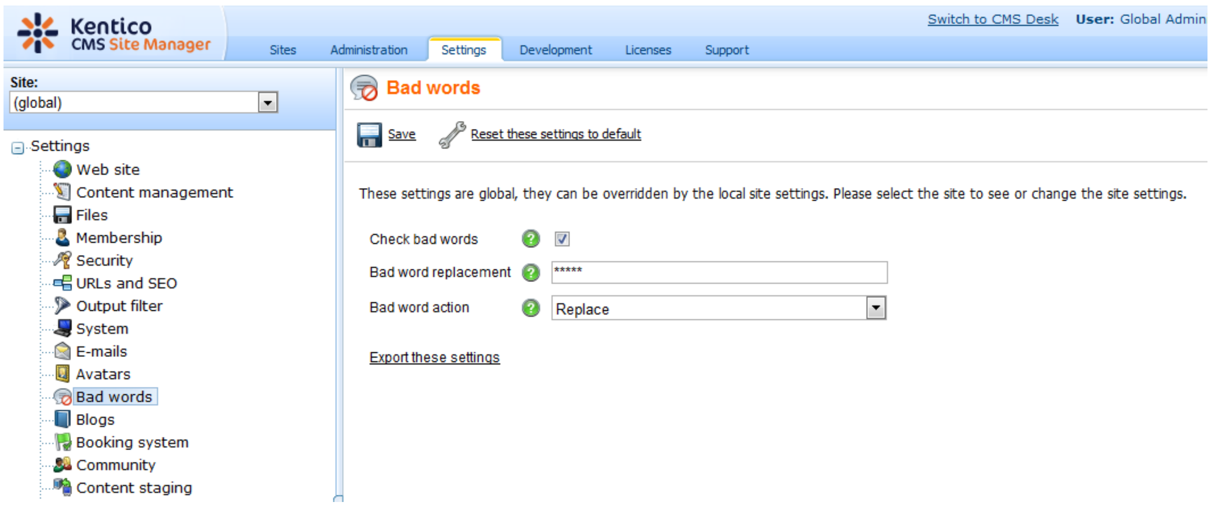
The Bad words module is just one of the modules which can help you deal with unwanted user input. Besides detecting unwanted input based on keywords, you may also let your site visitors report such content via the Abuse report module. The last resort in dealing with unwanted content may be blocking access to your site from a certain IP address. This can be achieved using the Banned IPs module.

8.6.2 Enabling the module

For the module to be functional, you first have to go to **Site Manager -> Settings -> Bad words** and enable the **Check bad words** option.

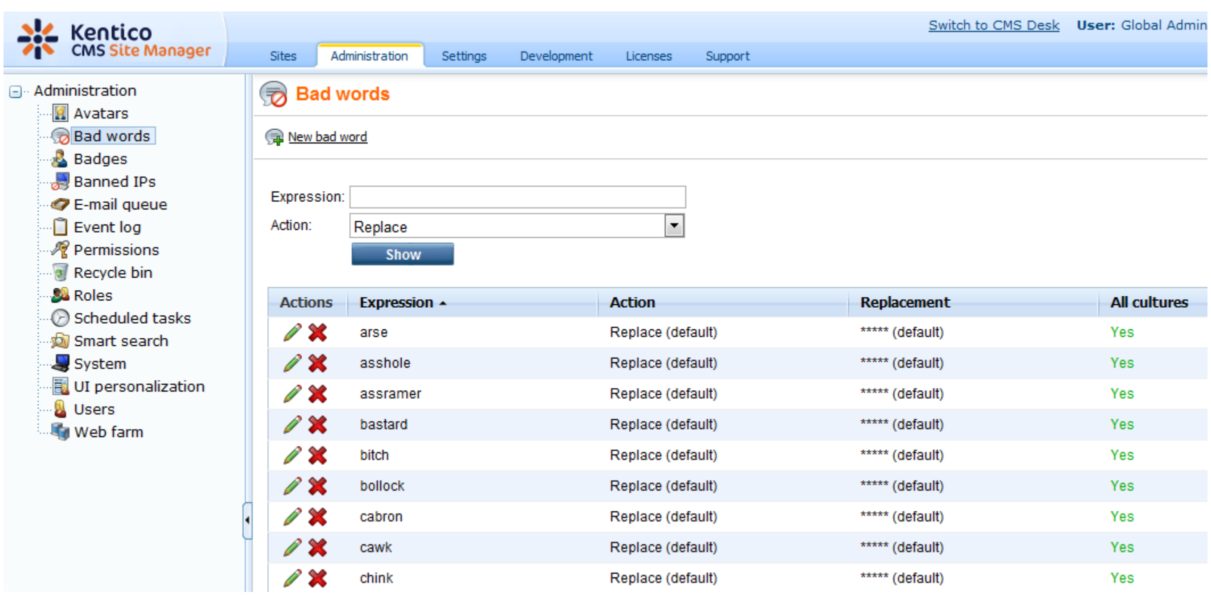
On this page, you can also define the default action that will be assigned to newly created bad words if the *Use default settings* option is enabled in the bad word creation dialog (see Defining a bad word for more details). The default action can be selected from the **Bad word action** drop-down list.

Default replacement string can be entered into the **Bad word replacement** field. It will be used in case that a bad word is detected, the *Replace* action should be performed on its detection, but has no replacement text defined (i.e. has the *Use default settings* option enabled for the *Replace by* field).



8.6.3 Defining a bad word

The Bad words module's user interface is located in **Site Manager -> Administration -> Bad words**. This is where all existing bad words are listed and where new bad words can be defined.



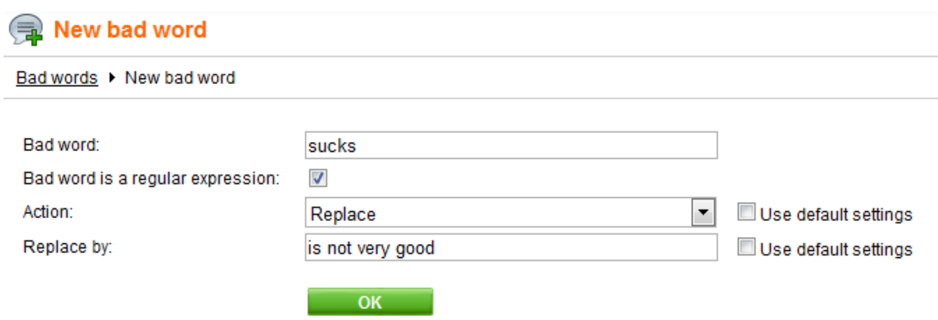
The top part of the page is a filter. Using it, you can display only those bad words that match the specified criteria. You can filter by the **Expression** and by the **Action** that should be taken on its detection. After specifying the filtering criteria and clicking the **Show** button, only those items that match the specified criteria will be displayed in the list.

Adding a new bad word

1. To add a new bad word, click the **New bad word** ( ) link at the top of the page. You will be redirected to the **New bad word** dialog. The following details can be entered:

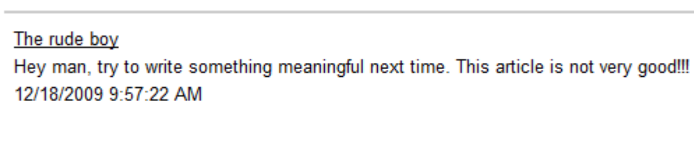
- **Bad word** - the string that should not appear in input text
- **Bad word is a regular expression** - if checked, the string entered into the previous field will searched as a regular expression
- **Action** - action that will be taken in case that the bad word is detected; see the Possible actions topic for more details
  Use default settings - if enabled, global value will be used as set in *Site Manager -> Settings -> Bad words -> Bad word action*
- **Replace with** - in case the Replace action is selected, the substitute for the bad word is defined here
  Use default settings - if enabled, global value will be used as set in *Site Manager -> Settings -> Bad words -> Bad word replacement*

Fill in the required details and click **OK**.

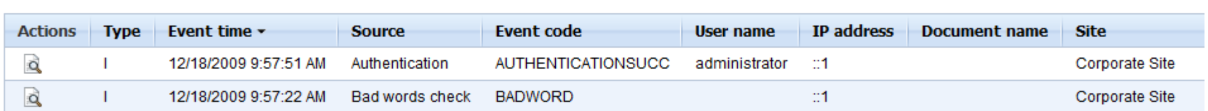


2. Let's try the functionality now. Go to the live Corporate Site, enter the Blog section and open some of the blog posts. Enter a comment as in the following screenshot and click **Add**.

3. As you can see, the last word has been replaced with its polite substitute that has been defined earlier.



4. You can also check the **Event log** in **Site Manager -> Administration**. An event is always logged automatically when user input containing a bad word is detected. The **Event code** is **BADWORD** in such case.



Actions	Type	Event time	Source	Event code	User name	IP address	Document name	Site
	I	12/18/2009 9:57:51 AM	Authentication	AUTHENTICATIONSUCC	administrator	::1		Corporate Site
	I	12/18/2009 9:57:22 AM	Bad words check	BADWORD		::1		Corporate Site

8.6.4 Possible actions

In case that a bad word is detected, one of the following actions can be taken:

- **Remove** - the bad word is removed from the entered text with no substitution
- **Replace** - the bad word is removed and replaced with a predefined string
- **Report abuse** - a report is created in **CMSDesk -> Tools -> Abuse report**
- **Request moderation** - the post is submitted to approval before being published; this happens even in case that the forum, message board or blog is not moderated by default
- **Deny** - a warning message will be displayed to the user when they try to post the inadequate text, listing which words should be removed

The following table shows which actions are available for each of the supported modules:

	Blog comments	Forums	Messaging	Message boards
Remove	•	•	•	•

Replace	•	•	•	•
Report abuse	•	•		•
Request moderation	•	•		•
Deny	•	•	•	•

In case that there is **more than one bad word detected** in input text and the words have **different actions** set, the actions will be taken according to their hierarchy. *Remove*, *Replace* and *Report abuse* actions are independent, i.e. the actions can be taken simultaneously. *Request moderation* is stronger than *Report abuse* and *Deny* is the strongest of all.

The following list shows which actions will be taken in some specific cases in order for you to easily understand the actions hierarchy:

- **Remove and Replace** - the first one will be removed and the second one will be replaced with the defined substitute
- **Remove, Replace and Report abuse** - the first one will be removed, the second one replaced and an abuse report will be logged
- **Remove, Replace and Request moderation** - the first one will be removed, the second one will be replaced and the post will have to be approved
- **Report abuse and Request moderation** - the post will have to be approved and no abuse report will be logged
- **Deny and any other** - the text is left as it is, but produces an error message

8.6.5 Multilingual support

You can define in which cultures will a certain bad word be filtered. If you choose to **Edit** ( ) a bad word in the list in **Site Manager -> Administration -> Bad words** and switch to its **Cultures** tab (the tab is not available in the **New bad word** dialog), you will be offered with the following two options:

- **The word is not allowed in all cultures** - the bad word will be filtered in all website cultures
- **The word is not allowed only in following cultures** - the bad word will be filtered only in cultures added to the list below

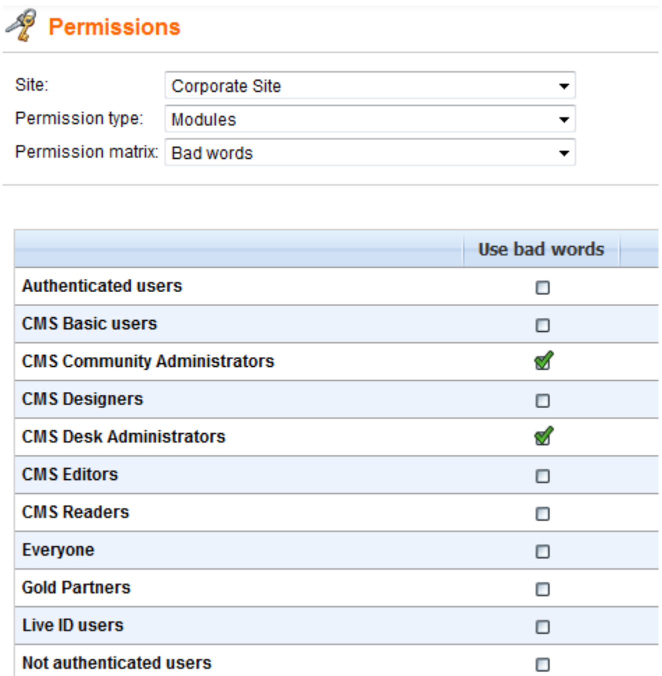
Using the **Add cultures** button, you can add cultures to the list. The **Remove selected button** removes all cultures selected by the check-boxes from the list.

8.6.6 Security

The Bad words module has only one permission to grant to roles:

- **Use bad words** - allows members of the role to use bad words, i.e. bad words filtering will not be performed to submissions made by members of the role

This permission can be granted to particular roles in **Site Manager -> Administration -> Permissions**, after selecting the **Modules -> Bad words** permission matrix.



8.6.7 Bad words internals and API

8.6.7.1 Database tables and API classes

The Bad words module uses the following database tables:

- **BadWords_Word** - this is the database table used for storing defined bad words
- **BadWords_WordCulture** - this database table is used for assigning bad words to website cultures

The Bad words API is provided by the following **CMS.SiteProvider** namespace classes:

- **BadWordInfo**, **BadWordInfoProvider** - these classes provide functionality for managing bad words
- **BadWordCultureInfo**, **BadWordCultureInfoProvider** - these classes provide functionality for managing assignments of bad words to cultures

The following topics show how methods from these classes can be used to manage bad words:

- Creating bad words
- Getting and updating bad words
- Deleting bad words
- Performing bad word checks

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.6.7.2 Creating bad words

The following code example shows how you can define a new bad word. The bad word will be checked in all website cultures.

[C#]

```
using System;
using CMS.SiteProvider;

...

        // Create a new bad word
        BadWordInfo bwi = new BadWordInfo();

        bwi.WordExpression = "ass";
        bwi.WordAction = BadWordActionEnum.Replace;
        bwi.WordReplacement = "xxx";
        bwi.WordIsGlobal = true;
        bwi.WordIsRegularExpression = false;
        bwi.WordLastModified = DateTime.Now;

        // Save the object to the database
        BadWordInfoProvider.SetBadWordInfo(bwi);
```

This second code example demonstrates how a culture-specific bad word can be defined. The bad word defined in this code example will be checked only in the Czech (*cs-cz*) culture.

[C#]

```
using System;
using CMS.SiteProvider;

...

        // Create a new bad word
        BadWordInfo bwi = new BadWordInfo();

        bwi.WordExpression = "prdel";
        bwi.WordAction = BadWordActionEnum.Replace;
        bwi.WordReplacement = "xxx";
        bwi.WordIsGlobal = false;
        bwi.WordIsRegularExpression = false;
```

```
        bwi.WordLastModified = DateTime.Now;

        // Save the object to the database
        BadWordInfoProvider.SetBadWordInfo(bwi);

        // Get the culture where the bad word should be checked
        CultureInfo ci = CultureInfoProvider.GetCultureInfo("cs-cz");

        // Create a new bad word culture object
        // for assigning the bad word to specified culture
        BadWordCultureInfo bwci = new BadWordCultureInfo();

        bwci.CultureID = ci.CultureID;
        bwci.WordID = bwi.WordID;

        // Save the object to the database
        BadWordCultureInfoProvider.SetBadWordCultureInfo(bwci);
```

8.6.7.3 Getting and updating bad words

The following code example demonstrates how to get a *DataSet* with all global bad words stored in the system database. Using the WHERE condition, you can specify which particular bad words you want to retrieve.

[C#]

```
using System.Data;
using CMS.SiteProvider;

...

        // Set the WHERE condition
        string where = "WordIsGlobal = 1";
        // Set the ORDER BY clause
        string orderBy = "WordExpression";

        // Get a DataSet of Bad word info objects according to the given parameters
        DataSet ds = BadWordInfoProvider.GetBadWords(where, orderBy);

        // Get a DataSet of Bad word info objects for given culture by its ID
        DataSet ds2 = BadWordInfoProvider.GetBadWords(1);
```

In the code example below, you can see how a single bad word can be retrieved from the system database, updated with some changes and saved back to the database.

[C#]

```
using CMS.SiteProvider;
using CMS.CMSHelper;
```

```
...

        // First get the bad word info by its ID
        BadWordInfo bwi = BadWordInfoProvider.GetBadWordInfo(1);

        // Set the WordReplacement property to the default value from the current si
        bwi.WordReplacement = BadWordInfoProvider.GetReplacementFromSettings(CMSCont

        // Save object
        BadWordInfoProvider.SetBadWordInfo(bwi);
```

8.6.7.4 Deleting bad words

This sample code explains how a single bad word can be deleted from the system database.

[C#]

```
using CMS.SiteProvider;

...

        // First get the bad word info by its ID
        BadWordInfo bwi = BadWordInfoProvider.GetBadWordInfo(1);

        if (bwi != null)
        {
            // Delete the bad word
            BadWordInfoProvider.DeleteBadWordInfo(bwi);
        }
```

8.6.7.5 Performing bad word checks

The first sample code on this page illustrates how a bad words check can be performed to a message sent via the Messaging module.

[C#]

```
using CMS.SiteProvider;
using CMS.CMSHelper;
using CMS.Messaging;

...

        // Get a message object by its ID
        MessageInfo message = MessageInfoProvider.GetMessageInfo(1);

        // Set columns for bad words check
        string columns = "MessageSubject;MessageBody";
```

```
        // Check if the message contains any bad word, perform appropriate bad word
        // and return eventual error text
        string errorText = BadWordsHelper.CheckBadWords(message, columns, CMSContext

        // Save message object
        MessageInfoProvider.SetMessageInfo(message);
```

The second code example shows how a bad words check can be performed to a custom string.

[C#]

```
using System.Collections;
using CMS.SiteProvider;
using CMS.CMSHelper;

...

        // Check if the current user can use bad words
        if (!BadWordInfoProvider.CanUseBadWords(CMSContext.CurrentUser, CMSContext.C
        {
            string cultureCode = "en-us";
            string text = "This is a string containing two bad words, namely asshole

            // Hashtable that will contain found bad words
            Hashtable foundWords = new Hashtable();

            // Modify the string according to found bad words and return which
action should be performed
            BadWordActionEnum action = BadWordInfoProvider.CheckAllBadWords
(cultureCode, CMSContext.CurrentSiteName, ref text, foundWords);

            switch (action)
            {
                case BadWordActionEnum.Deny:
                    // Some additional actions performed here ...
                    break;

                case BadWordActionEnum.RequestModeration:
                    // Some additional actions performed here ...
                    break;

                case BadWordActionEnum.Remove:
                    // Some additional actions performed here ...
                    break;

                case BadWordActionEnum.Replace:
                    // Some additional actions performed here ...
                    break;

                case BadWordActionEnum.ReportAbuse:
                    // Some additional actions performed here ...
```

```
                    break;

                case BadWordActionEnum.None:
                    // Some additional actions performed here ...
                    break;
            }
        }
```

8.7 Banned IPs

8.7.1 Overview

The Banned IPs module is useful when you want to prevent users with certain IP addresses from accessing or using your website in a certain way. This typically happens when a user posts offensive material on a website (e.g. on a forum), harasses site members or behaves in some other unwanted way. IP banning can also be used to restrict access to your websites from certain areas of the world. These bans can be set either for individual websites or globally for all websites in the system.

To learn how to enable the Banned IPs module, please see the Enabling IP banning topic.

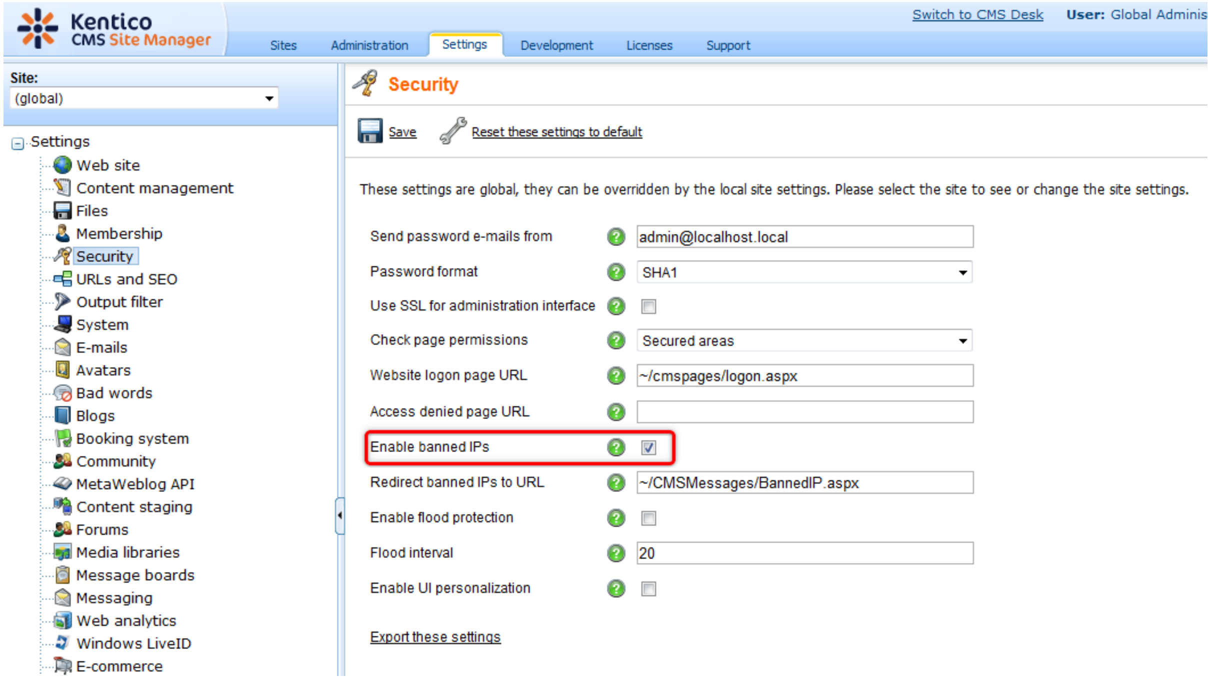
Adding banned IPs can easily be performed by site administrators. The Banning an IP address topic describes how. To find out how to determine which IP a user you want to ban is using, please see the Finding an IP address topic.

The Banned IPs internals and API sub-chapter provides information about the database tables and classes used by the module and examples of how banned IPs can be managed using the API.

IP banning is usually used as a last resort and there are other modules which can help you deal with offensive content and keep your website clean. Please refer to the Modules -> Abuse report and Modules -> Bad words chapters of this guide to read more. If you only wish to temporarily kick a user from a website, the On-line users module provides a way to do so.

8.7.2 Enabling IP banning

For the bans to take effect, go to **Site Manager -> Settings -> Security**, select the appropriate site, check the **Banned IP enabled** check-box and click **Save**. The bans should take effect now.



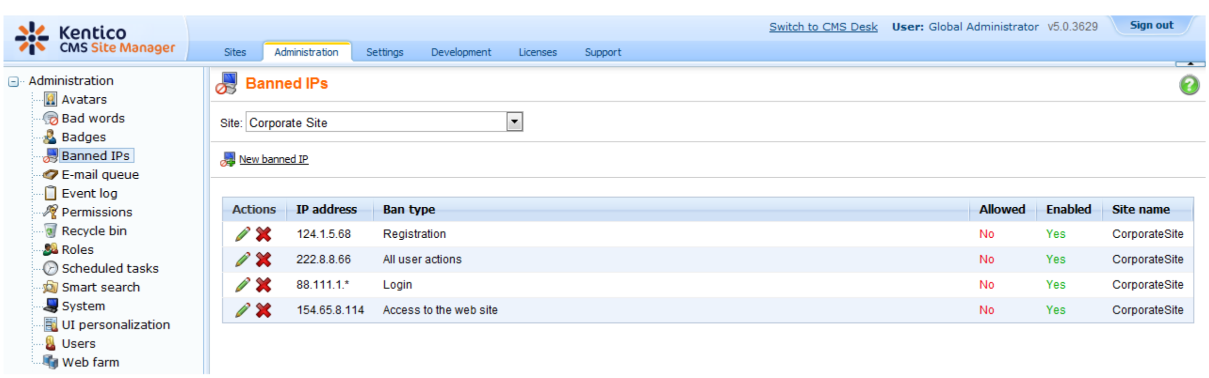
Users attempting to perform an action from an IP address that is banned should get a page with the following message displayed in their browsers.

This is the default banned IPs redirect page, which can be found at *~/CMSMessages/BannedIP.aspx*. You can create your own page and set its URL in **Site Manager -> Settings -> Security -> Redirect banned IPs to URL**.

8.7.3 Banning an IP address

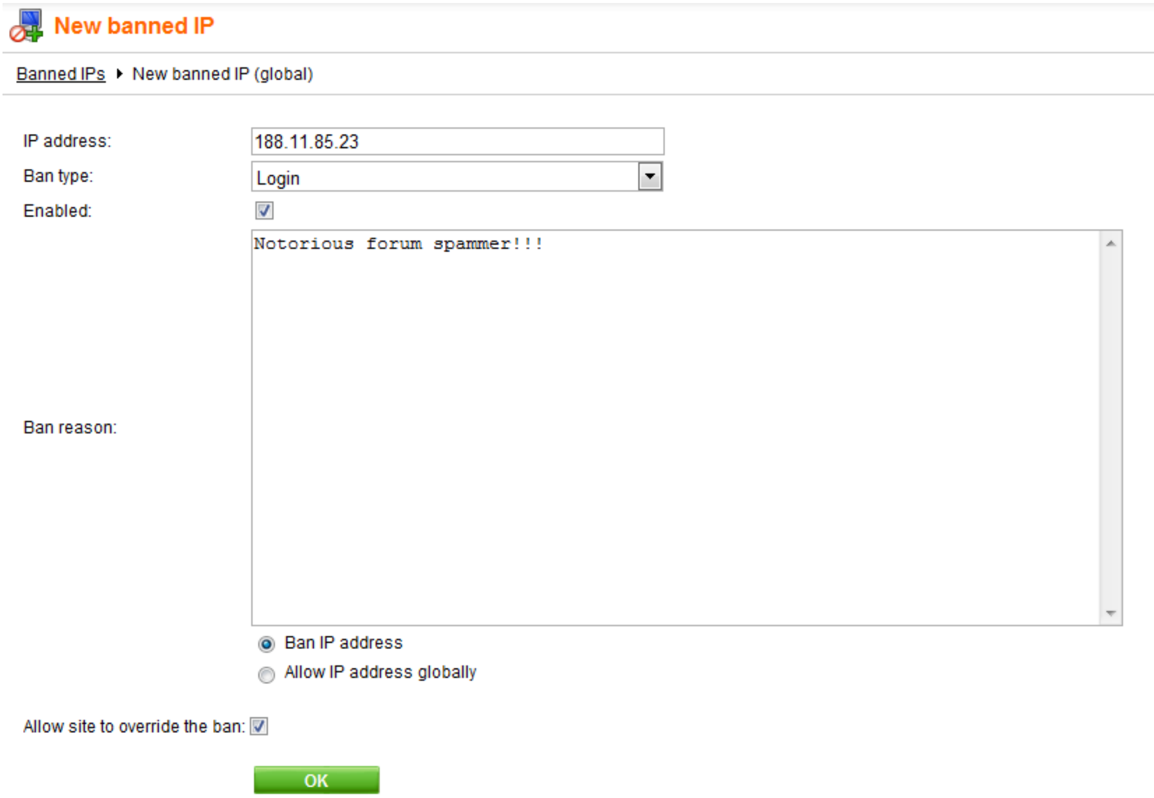
Banned IPs can be managed in **Site Manager -> Administration -> Banned IPs**. Alternatively, you can access the same page from **CMS Desk -> Administration**, but you can manage banned IPs only for the currently edited site.

From the **Site** drop-down list, choose the site you want to add the IP for and click the **New banned IP** link.



Enter the required details:

IP Address	IP address to be banned. Asterisk (*) can be used as a substitute for any number in the IP address, substituting for all values from 0 to 255. Example: 192.168.1.*
Ban type	Type of the ban. The following types are available: • Access to the website - users cannot access the site from the entered address at all • Login - users cannot log in from the entered address • Registration - users cannot register from the entered address • All user actions - users can enter the site from this IP, but specific actions will not be allowed
Enabled	If unchecked, the ban takes no effect.
Ban Reason	Text description of why the IP was banned.
Ban IP address	If selected, this IP address will be banned.
Allow IP address for this site if the IP address is banned globally	If selected, this IP address will be allowed for the selected site even if it is banned globally.
Allow site to override the ban	If checked, this ban can be overridden by bans set for particular sites; can be set only for global bans.



When you have all details entered, click **OK**.

Now if you go back to the list of banned IPs, you should see the newly created record present in the list.

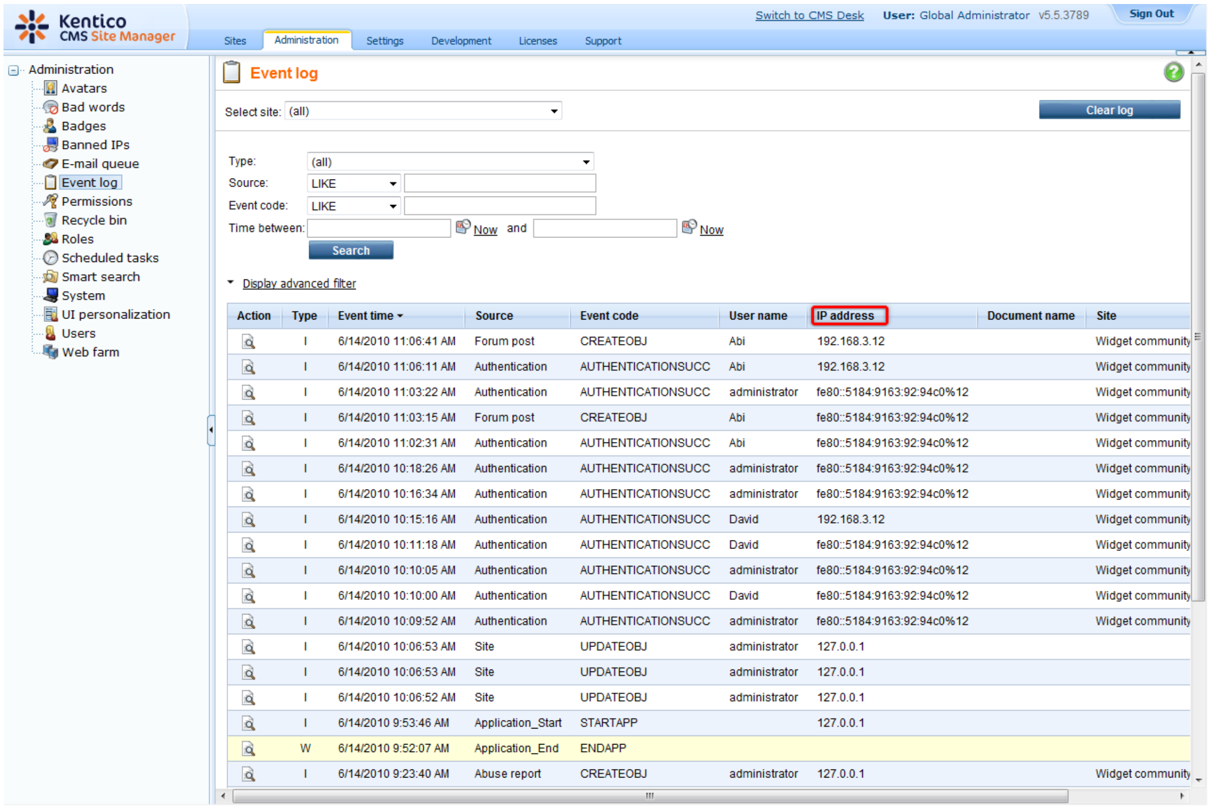
Editing an existing ban

If you want to edit the record in the future, just go to the list and click the **Edit** () icon next to the record. The same page will be displayed as when creating a new record, but with details entered. To make the desired changes, just change the appropriate values and click **OK**.

8.7.4 Finding an IP address

There are several ways to locate user IP addresses.

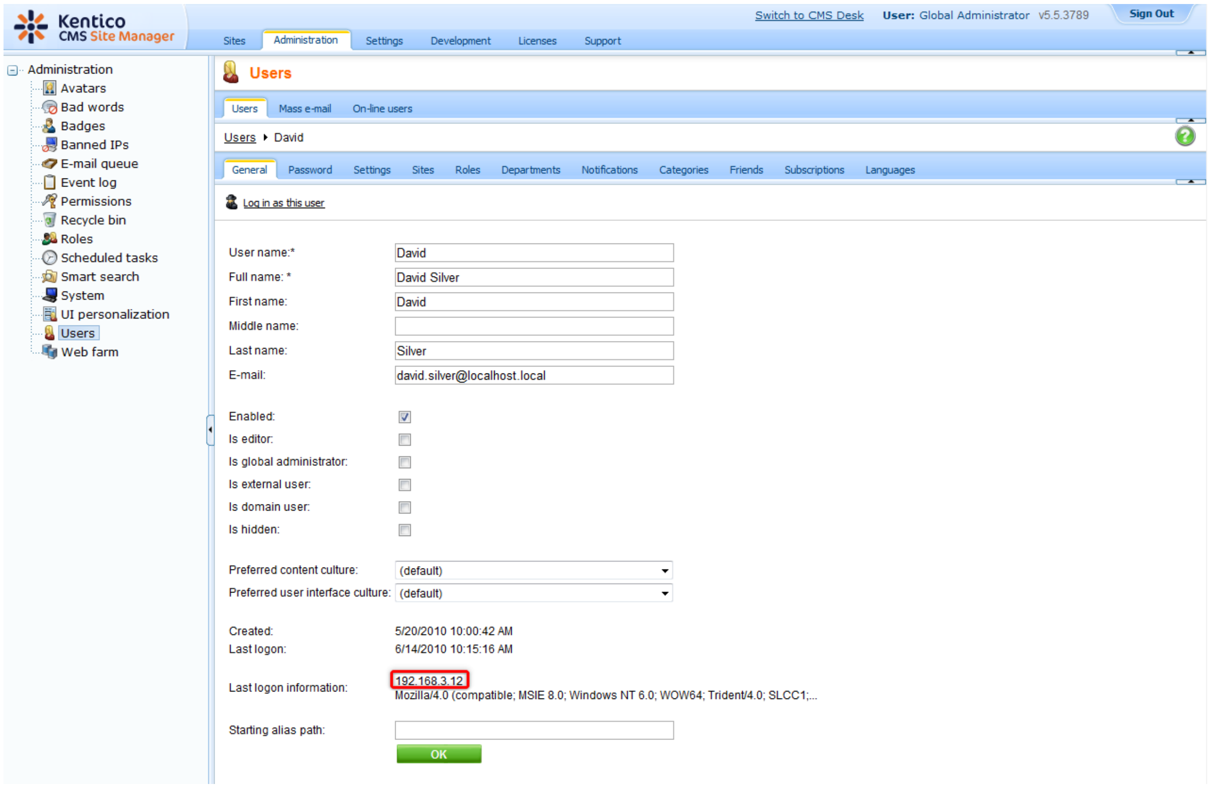
If you wish to find the IP address of a user responsible for a certain event, go to **Site Manager -> Administration -> Event log** and find the event you are interested in. The IP address of the user who caused the event has its own column in the displayed table and can also be viewed if you choose to click **Display event** (). This can also be done at **CMS Desk -> Administration -> Event Log**, but here you can only view events from the currently edited site.



For example, if you want to find a user who often uses Bad words, enter *BADWORD* into the **Event code** field of the Event log filter and make sure **LIKE** is selected from the drop-down list. This will display a list of all Bad word violations and all relevant information including user names and their IP addresses.

To find the IP address used by a specific user when they last logged on, go to **Site Manager -> Administration -> Users**. Alternatively, you can access this information from **CMS Desk -> Administration -> Users**, but you can only see the users of the currently edited site.

Choose to **Edit** () the user whose IP address you wish to find. It can be seen under **General -> Last logon information**. This can be useful if you get multiple abuse reports about some user and want to quickly find their IP address.

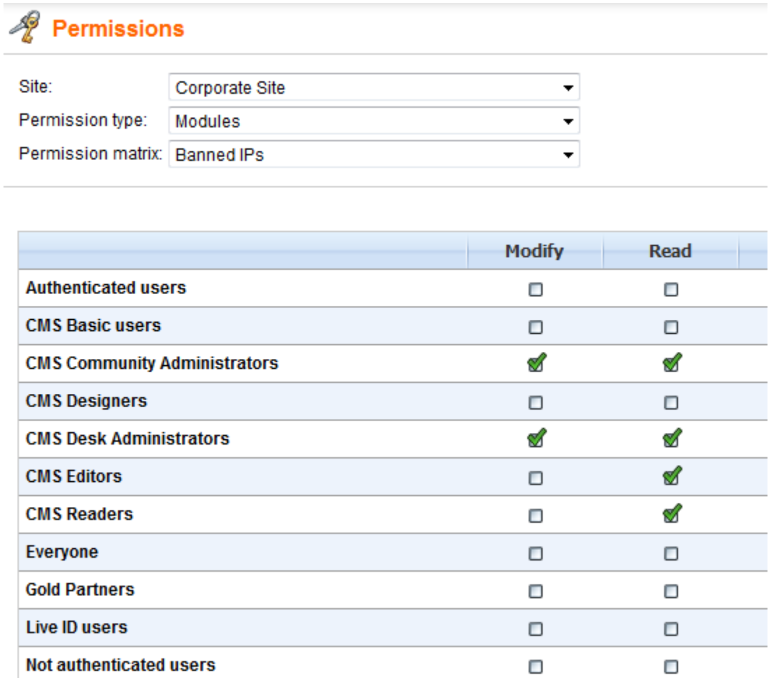


8.7.5 Security

To limit access to the Banned IPs module, go to **Site Manager -> Administration -> Permissions** and grant roles with appropriate permissions according to your needs.

The following permissions can be assigned to the roles:

- **Modify -** members of the role are allowed to add, edit and delete banned IPs
- **Read -** members of the role are allowed to view the banned IPs list



	Modify	Read
Authenticated users	☐	☐
CMS Basic users	☐	☐
CMS Community Administrators	✓	✓
CMS Designers	☐	☐
CMS Desk Administrators	✓	✓
CMS Editors	☐	✓
CMS Readers	☐	✓
Everyone	☐	☐
Gold Partners	☐	☐
Live ID users	☐	☐
Not authenticated users	☐	☐

8.7.6 Banned IPs internals and API

8.7.6.1 Database tables and API classes

The Banned IPs module uses the following database table:

- **CMS_BannedIP** - this is the database table used for storing banned IPs

The Banned IPs API is provided by the following **CMS.SiteProvider** namespace classes:

- **BannedIPInfo**, **BannedIPInfoProvider** - these classes provide functionality for managing banned IPs

The following topics show examples of how these classes can be used:

- Getting banned IP data
- Modifying and creating banned IPs
- Checking if actions are banned

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.7.6.2 Getting banned IP data

The following sample code shows how you can get a banned IP as a *BannedIPInfo* object:

[C#]

```
using CMS.SiteProvider;

...

        // Get BannedIP object by IPAddressID
        BannedIPInfo bi = BannedIPInfoProvider.GetBannedIPInfo(1);
```

The following sample code shows how you can get a DataSet containing all banned IPs in the system with an *IPAddressID* greater than *5*, ordered by their *IPAddress*:

[C#]

```
using System.Data;
using CMS.SiteProvider;

...

        string where = "IPAddressID > 5";
        string orderBy = "IPAddress";

        // Get DataSet of BannedIP objects
        DataSet ds = BannedIPInfoProvider.GetBannedIPs(where, orderBy);
```

8.7.6.3 Modifying and creating banned IPs

The following sample code shows how you can modify the Ban reason of an existing banned IP:

[C#]

```
using CMS.SiteProvider;

...

        // Get BannedIP object by IPAddressID
        BannedIPInfo bi = BannedIPInfoProvider.GetBannedIPInfo(1);

        // Change Ban reason property
        bi.IPAddressBanReason = bi.IPAddressBanReason + "_Updated";

        // Save object
        BannedIPInfoProvider.SetBannedIPInfo(bi);
```

The following sample code shows how you can create a new banned IP of type *All user actions* and add it to the system:

[C#]

```
using CMS.SiteProvider;

...

        // Create new BannedIP object
        BannedIPInfo bi = new BannedIPInfo();

        // Set properties
        bi.IPAddress = "127.0.0.1";
        bi.IPAddressBanReason = "Ban reason";
        bi.IPAddressAllowed = true;
        bi.IPAddressAllowOverride = true;
        bi.IPAddressBanType = BannedIPInfoProvider.BanControlEnumString(BanControlEn
        bi.IPAddressBanEnabled = true;

        // Save object
        BannedIPInfoProvider.SetBannedIPInfo(bi);
```

8.7.6.4 Checking if actions are banned

The following sample code shows how you can check if the current user action is banned for the current site. It uses a Label control with the ID *lblError* to display the error message:

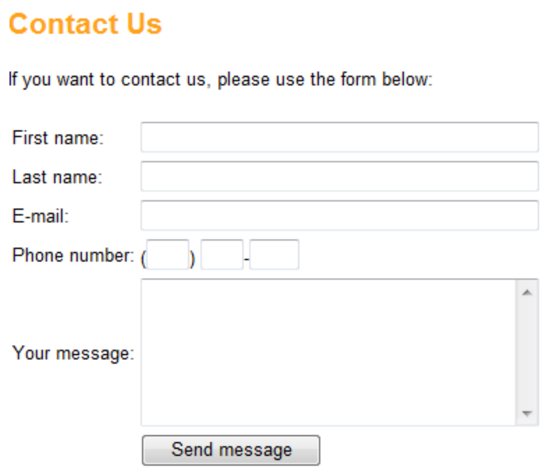
[C#]

```
using CMS.SiteProvider;
using CMS.CMSHelper;
using CMS.GlobalHelper;

...

        // Check if current action is allowed
        if (!BannedIPInfoProvider.IsAllowed(CMSContext.CurrentSiteName, BanControlEn
        {
            // If not error message is shown
            lblError.CssClass = "ErrorLabel";
            lblError.Text = "Requested action was canceled. Banned IP address.";
        }
```

8.8 BizForms

8.8.1 Overview

The BizForms module enables non-technical users (content editors) to easily create and publish on-line forms. These forms can be used to gather structured data from website users. A typical example of such form is the **Contact Us** form, which can be found on the **Company** page of the sample Corporate Site. The form is depicted in the screenshot below.



Administration interface of the module is located in **CMS Desk -> Tools -> BizForms**. This is where new BizForms can be created, as described in the Creating a new form topic. Once a BizForm is created via the user interface, it can be added to a live site page. This can be done either via the WYSIWYG editor, or by means of the **BizForm (on-line form)** web part or widget. Both options are described in the Displaying a form on the live site topic.

Each BizForm has its own separate database table where submitted data is stored. The submitted data can be viewed and managed in **CMS Desk -> Tools -> BizForms**. Data records can even be exported to a **Microsoft Excel** spreadsheet in this part of the user interface. Please refer to the Managing BizForm data topic for more information on which actions can be performed with the submitted data.

You may want to notify the person who is responsible for management of BizForms data when a new record is submitted. For this purpose, you can configure the form to send automatic notification e-mails about new records. The user who submitted the new record may also be notified that the record was submitted successfully and that it will be processed soon. This can be achieved by configuring autoresponder e-mails. Both options are described in the Notification and autoresponder e-mails sub-chapter. Macros can be used largely with notification and autoresponder e-mails, as well as when you want to localize field captions in the form. Please refer to the Using macros with BizForms topic to learn more.

Access to BizForms may be restricted, so that only members of selected roles can perform certain operations with the forms themselves or with the data in them. The Security topic gives you an overview of how to configure BizForms permissions both globally and separately for each particular form.

Layout of each form is fully customizable. The Defining custom form layout topic explains how it can be achieved. You may also customize behavior of the BizForms module with your custom code. These possibilities are explained in the BizForms internals and API sub-chapter, along with an overview of where BizForm records are stored and which API classes contain methods for their management. It also provides a series of code examples showing how methods from these classes may be used in your custom code.

8.8.2 Creating a new form

In this topics, we will create a new sample BizForm via the BizForms module's user interface. This topic doesn't explain all options that are available in the user interface. For a detailed description of each option, please refer to the built-in context help, which is accessible by clicking the icon in the top-right corner of the user interface.

1. Go to **CMS Desk -> Tools -> BizForms** and click the **New form** link. Enter the following details:

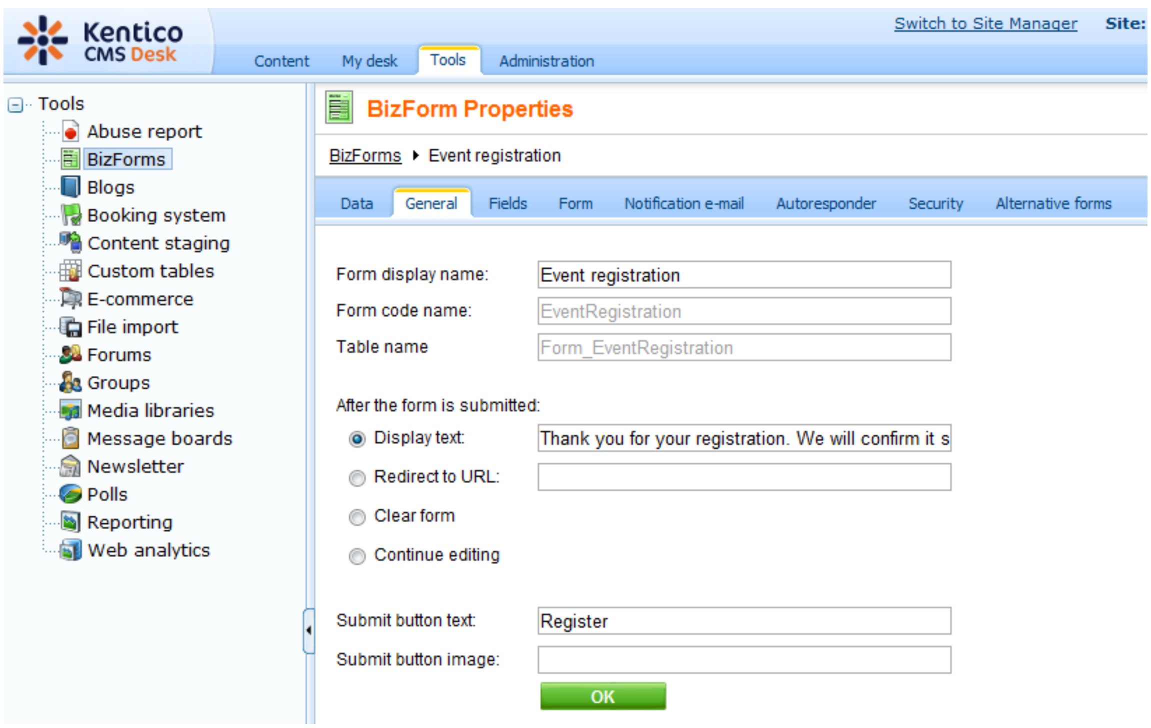
- **Form display name:** Event registration
- **Form code name:** EventRegistration
- **Table name:** Form_EventRegistration

Click **OK**.

2. You will be redirected to the **General** tab of the new BizForm's editing interface. Enter the following values:

- **Display text:** Thank you for your registration. We will confirm it shortly by e-mail.
- **Submit button text:** Register

Click **OK**.



3. Now we will define the form fields. Go to the **Fields** tab. Add the following fields using the **Add attribute** ( ) button. For each field, enter the values, click **OK** and repeat the procedure until you have all the listed fields defined.

- **Column name**: FirstName
- **Show on public form**: enabled
- **Field caption**: First name
- **Field type**: Text box
- **Maximum length**: 100
- **Allow empty value**: disabled

- **Column name**: LastName

- **Show on public form**: enabled
- **Field caption**: Last name
- **Field type**: TextBox
- **Maximum length**: 100
- **Allow empty value**: disabled

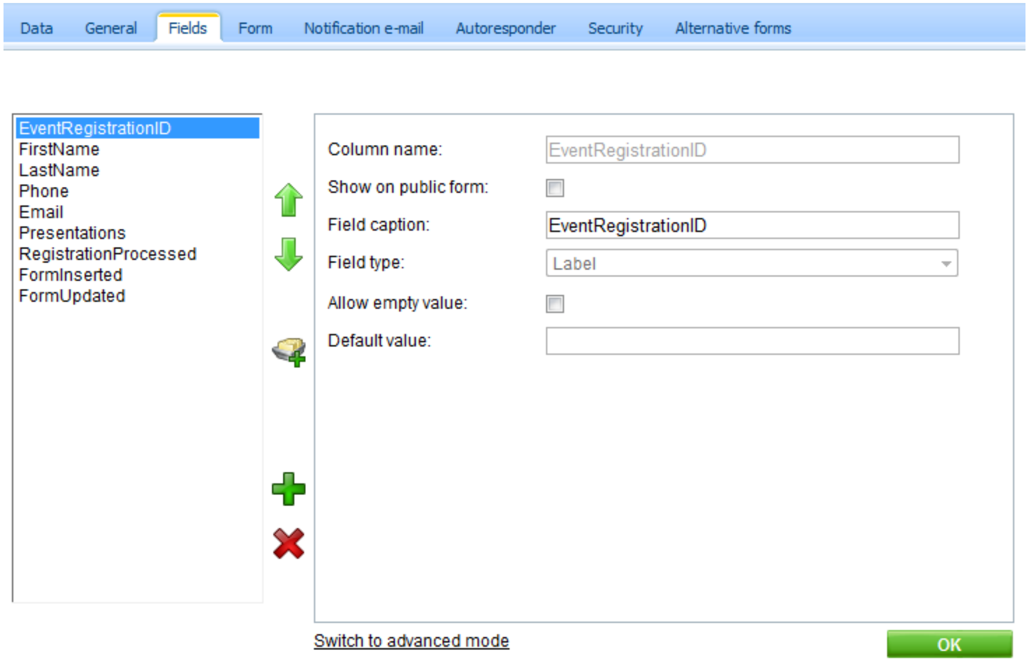
- **Column name**: Phone
- **Show on public form**: enabled
- **Field caption**: Phone
- **Field type**: U.S. phone number
- **Maximum length**: 14
- **Allow empty value**: enabled

- **Column name**: Email
- **Show on public form**: enabled
- **Field caption**: E-mail
- **Field type**: E-mail
- **Maximum length**: 100
- **Allow empty value**: disabled

- **Column name**: Presentations
- **Show on public form**: enabled
- **Field caption**: Presentations you want to visit
- **Field type**: Multiple choice
- **Options**: ASP.NET;ASP.NET
  ATLAS;ATLAS
  WPF;Windows Presentation Foundation
- **Allow empty value**: enabled

4. The last item will be used only by site editors to mark the processed registration forms. This will be done exclusively via the user interface. Therefore, it has the **Show on public form** property disabled.

- **Column name**: RegistrationProcessed
- **Show on public form**: disabled
- **Field caption**: Registration processed
- **Field type**: Check box
- **Allow empty value**: enabled



5. With all the fields defined, the form is ready to be published on the live site. This can be done several different ways, all of which are described in the [Displaying a form on the live site](#) topic.

Please note

If you click the **Switch to advanced mode** link below the field editor, extended user interface will be displayed. There you can configure additional options such as input validation rules or CSS classes used for the fields.

8.8.3 Displaying a form on the live site

In the Creating a new form topic, you have learned how to create a new BizForm via the module's user interface. This topic will give you an overview of how you can add the form to the live site.

Content editors can add a BizForm to any page with editable text regions. It can be done either using the **Insert BizForm** ( ) button on the WYSIWYG editor toolbar, or directly by typing a macro in the following format: *%%control:BizFormControl?BizFormCodeName%%*. The *BizFormCodeName* part of the macro needs to be replaced with the code name of the particular BizForm.

Website designers or developers can add a BizForm to any web part or widget zone by adding the **BizForm (on-line form)** web part or widget. In this case, code name of the BizForm needs to be entered in the **Form name** property of the web part or widget.

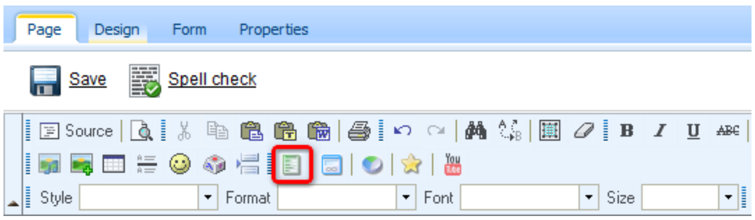
The examples below explain both options of adding BizForms to your live site pages.

Example 1: Adding a BizForm via the WYSIWYG editor

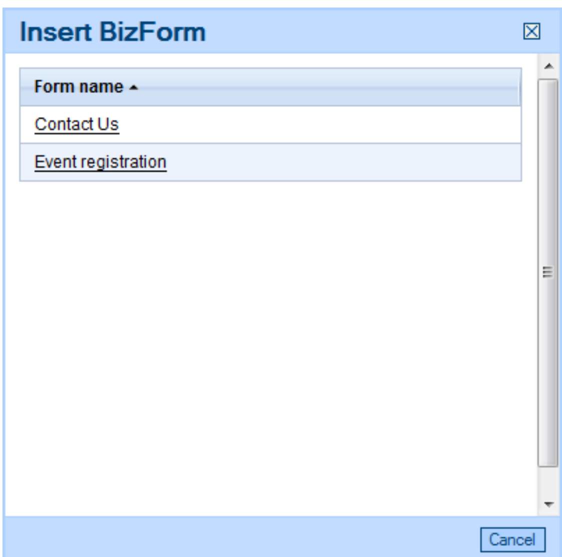
In this example, we will add the *Event registration* form, created in the [Creating a new form]() topic, to a page using the **Insert BizForm** () button on the WYSIWYG editor toolbar. Any other BizForm can be added to a page exactly the same way.

1. Go to **CMS Desk -> Content** and select some page with editable regions (the **Editable text** web part). View the page in the **Edit -> Page** mode.

2. Click **Insert BizForm** () button in the WYSIWYG editor toolbar.

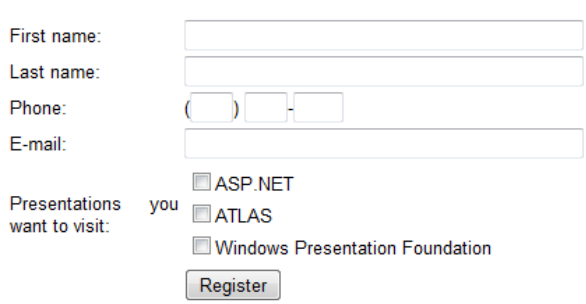


3. The Insert BizForm dialog opens. Click **Event registration** to insert the form into the text.



The following macro will be pasted into the text: *%%control:BizFormControl?EventRegistration%%.* Typing the macro into the text manually, without using the **Insert BizForm** () button, would lead to exactly the same result.

4. Click **Save** and switch to the **Live site** mode. You will see the form on the page.



5. Try entering some values and submitting the filled-in form. After doing that, you may proceed to the Managing BizForm data topic, where management of submitted data is explained.

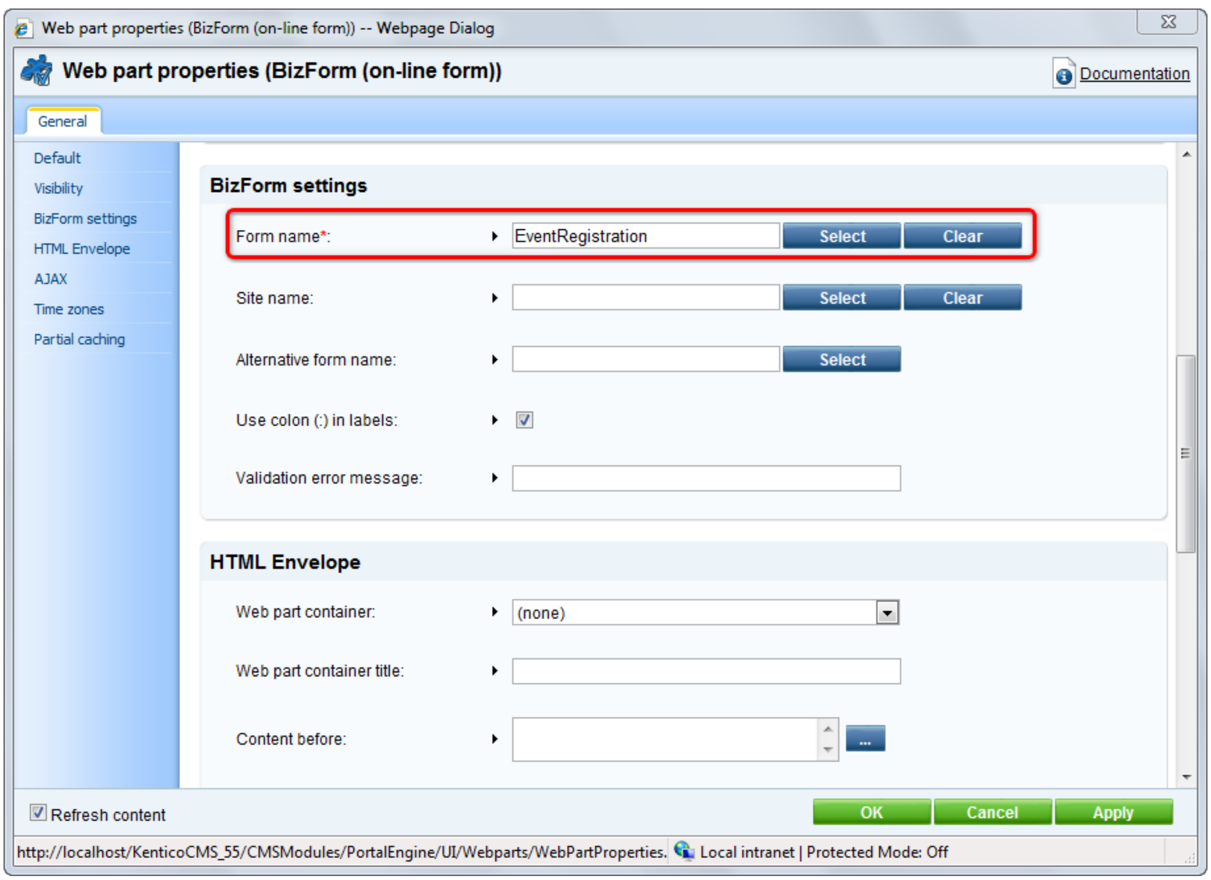
Example 2: Adding a BizForm using the BizForm (on-line form) web part

This example demonstrates how to add the *Event registration* form, created in the Creating a new form topic, to a page using the BizForm (on-line form) web part. Any other BizForm can be added to a page exactly the same way. The same applies to using the BizForm (on-line form) widget - the only difference is that the widget can only be added to a widget zone, not a web part zone.

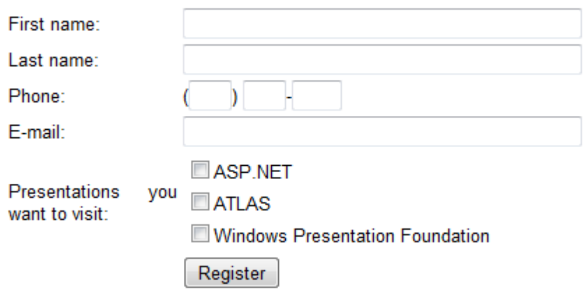
1. Go to **CMS Desk -> Content** and select some page with a web part zone. View the page in the **Edit -> Design** mode.

2. Click the **Add web part** () button at the top-right corner of the zone and choose the **BizForms -> BizForm (on-line form)** web part. Click **OK**.

3. In the web part properties dialog, you only need to enter *EventRegistration* (the code name of the form) into the **Form name** property and click **OK**.



4. Click **Save** and switch to the **Live site** mode. You will see the form on the page.



5. Try entering some values and submitting the filled-in form. After doing that, you may proceed to the Managing BizForm data topic, where management of submitted data is explained.

8.8.4 Managing BizForm data

BizForm records (data submitted by website users) can be viewed and managed in **CMS Desk -> Tools -> BizForms**. In the list of BizForms, you need to click the **Edit** ( ) icon next to a particular BizForm. You will get redirected to the **Data** tab of the BizForm's editing interface, where particular records are listed.

Each of the records listed in the table has the following options:

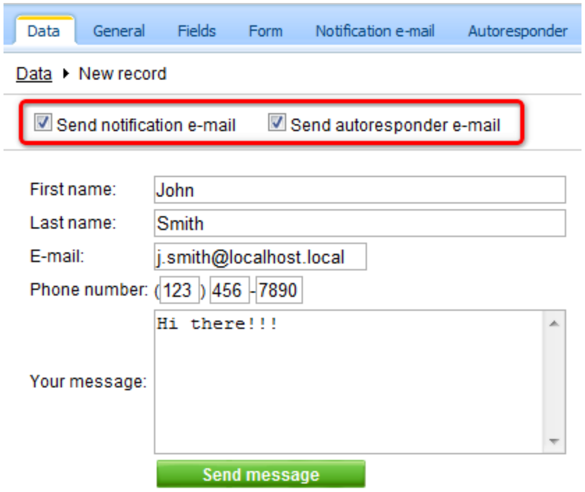
- **Edit record ( )** - allows you to alter the record
- **Delete record ( )** - allows you to delete the record

You can also perform the following actions using the links above the listing:

- **New record** - creates a new record in this BizForm
- **Select displayed fields** - enables you to tell which columns will be displayed in the listing
- **Export data to Excel** - exports all records into a new Excel spreadsheet (*.xls* file)
- **Export data to Excel 2007** - exports all records into a new Excel 2007 spreadsheet (*.xslx* file)

When editing ( ) or creating ( ) a record via the administration interface, the BizForm is displayed the same way as on the live site, letting you enter or change the values. On top of it, you can decide if notification and autoresponder e-mails will be sent when you save the record:

- **Send notification e-mail** - if enabled, notification e-mail will be sent when the record is saved (according to configuration on the **Notification e-mail** tab)
- **Send autoresponder e-mail** - if enabled, autoresponder e-mail will be sent when the record is saved (according to configuration on the **Autoresponder** tab)



8.8.5 Notification and autoresponder e-mails

8.8.5.1 Notification and autoresponder e-mails

The BizForms module allows you to send two types of e-mails automatically when a new record is added:

- Notification e-mail - e-mail notifying the person responsible for BizForms data management (content editor, administrator, ...) about the new record
- Autoresponder - e-mail to the person who submitted the new record, typically confirming that the record has been received and will be processed

When a record is added on the live site, the e-mails are sent out based on settings described in the following topics. When a record is added or modified via the administration interface, the person who is entering the record can decide whether these e-mails will be sent, as described in Managing BizForm data.

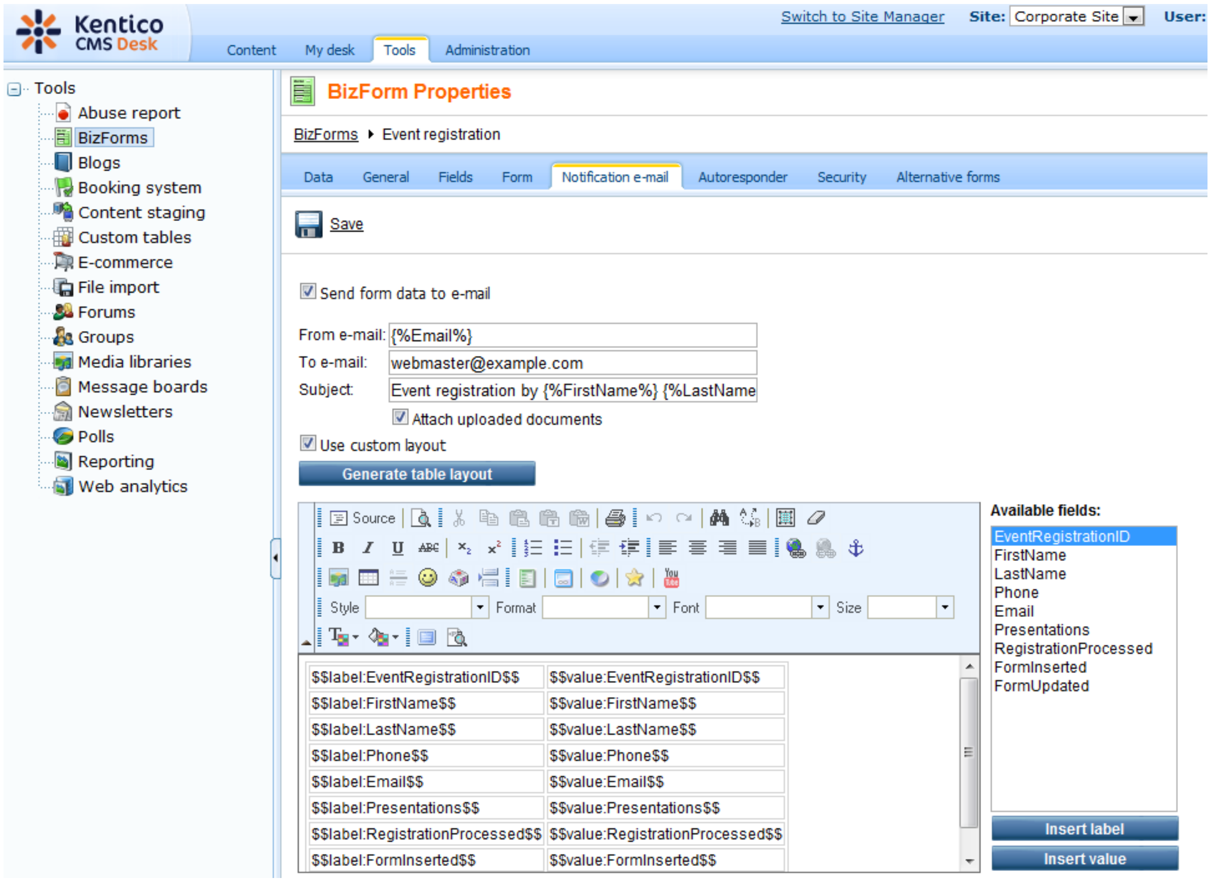
8.8.5.2 Notification e-mails

Notification e-mails can be configured on the **Notification e-mail** tab of a form's editing interface. To enable them, you first need to enable the **Send form data to e-mail** option. Then you need to configure the following options:

- **From e-mail -** e-mail address of the user who submitted the form; you will typically use a macro to get a value from a field of the form where users enter their e-mail address, as described in the Using macros with BizForms topic
- **To e-mail -** e-mail address where notification e-mails should be sent; typically address of the person responsible for management of BizForm records
- **Subject** - subject of the notification e-mails
- **Attach uploaded documents** - enable this option if you want to attach documents submitted via the form (if there are some) to the e-mails
- **Use custom layout** - if disabled, notification e-mails' body will contain all field names with the entered values, each on a single line; if enabled, the text area is displayed, letting you define custom layout of the e-mail body
- **Generate table layout** - this button generates a table with all field names in the left column and their values in the right column; macros for particular field names and values can also be entered

separately by selecting a field from the **Available fields** listbox and clicking the **Insert label** or **Insert value** buttons

Macros may be utilized in values of these fields, as described in the Using macros with BizForms topic. With all the options configured, click **Save** to save your configuration.



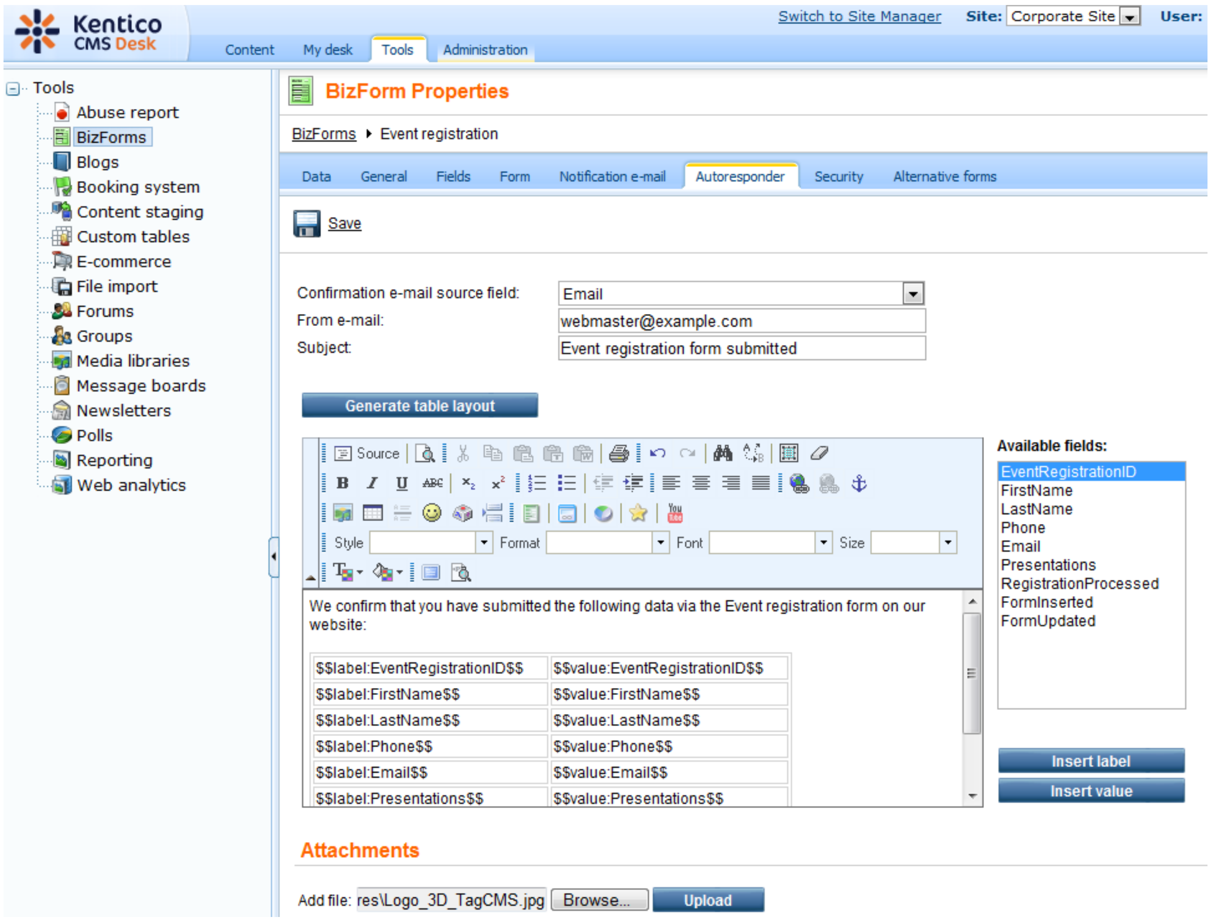
8.8.5.3 Autoresponder e-mails

Autoresponder e-mails can be configured on the **Autoresponder** tab of a BizForm's editing interface. When you navigate to the tab for the first time, just the **Confirmation e-mail source field** drop-down list will be visible. Expand it and choose the field which contains the e-mail address that the automatic response should be sent to. This means that autoresponder e-mails can only be used with forms where such a field is present (however, this is the case of most real-world on-line forms). After choosing the field, the following extra options will be displayed:

- **From e-mail** - e-mail address of the sender of the autoresponder e-mails (the *From* field of the e-mail message)
- **Subject** - subject of the autoresponder e-mails
- **E-mail body** - the main text area allows you to define the body of the autoresponder e-mails
- **Attachments** - this option allows you to attach a file to each autoresponder e-mail, such as event agenda, white papers, etc.
- **Generate table layout** - this button generates a table with all field names in the left column and their values in the right column; macros for particular field names and values can also be entered separately by selecting a field from the **Available fields** listbox and clicking the **Insert label** or

Insert value buttons

Macros may be utilized in values of these fields, as described in the Using macros with BizForms topic. With all the options configured, click **Save** to save your configuration.



8.8.6 Using macros with BizForms

Values of the fields that you need to fill in when configuring a BizForm can be obtained dynamically from the submitted forms. The following table shows which macro types can be used for particular values of BizForm configuration.

	Localization	Context	QueryString	Cookie	Custom	Path
Fields tab - **Default value** attribute	•	•	•	•	•	•
Notification e-mail and **Autoresponder** tabs - all fields	•	•	•	•	•	
General tab - **Form display name, Display text , Submit button text; Fields**	•					

tab - **Field caption; Form** tab - **Table layout**						
General tab - Redirect to URL		•				

Context (data) macros

You can use values from current context when configuring a BizForm. This can be achieved using a data macro in the **{%column_name%}** format. The *column_name* part of the macro is the value of the **Column name** property of a particular BizForm filed. When the form is submitted, the macros are automatically resolved and replaced with particular data from the form.

These macros can be used in the following fields:

- **General tab**
  - Display text
  - Redirect to URL
- **Notification e-mails**
  - From e-mail
  - To e-mail
  - Subject
  - E-mail body
- **Autoresponder e-mails**
  - From e-mail
  - Subject
  - E-mail body

Example 1

When configuring notification e-mails for the sample BizForm created in the Creating a new form topic, you may use the following values:

- **From e-mail:** *{%Email%}*;
- **Subject:** *Event registration by {%FirstName%} {%LastName%}*;

This will result in the e-mail having the user's e-mail address as the sender address, which enables the recipient to easily reply to the e-mail. The subject of the e-mail will have the first and last name of the user, which will help identifying the sender of the e-mail.

Example 2

If you entered the **Display text** value on the **General** tab of the same form like this:

- *Dear {%FirstName%}, thank you for your message. We will contact you shortly.*

The text will be resolved as the following when "Jane" was entered in the **First name** field by the user:

- *Dear Jane, thank you for your message. We will contact you shortly.*

Form text localization

If you need to display the form on a multi-lingual website, you can localize field captions and other text strings using localization macros, e.g.:

- {$myform.fullname$}
- {$=Hello|de-de=Hallo|it-it=Ciao$}

You can find more details in Development -> Multilingual and international support -> Localization expressions.
Detailed overview of all macros in Kentico CMS can be found in Appendix A - Macro expressions.

8.8.7 Defining custom form layout

If you're not satisfied with the standard table layout of the form, the BizForms module allows you to define custom form layout. This can be achieved on the **Form** tab of a BizForm's editing interface in **CMS Desk -> Tools -> BizForms**.

Example

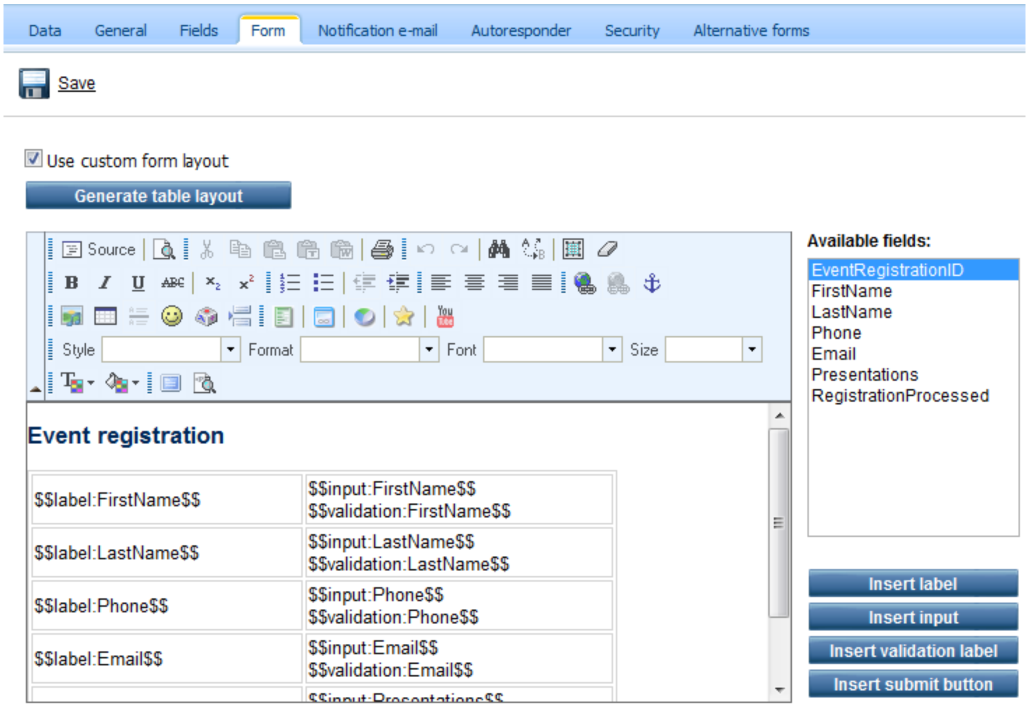
To modify the default layout of the sample *Event registration* form created in the Creating a new form topic, please take the following steps:

1. Go to **CMS Desk -> Tools -> BizForms** and click the **Edit** () icon next to the BizForm.

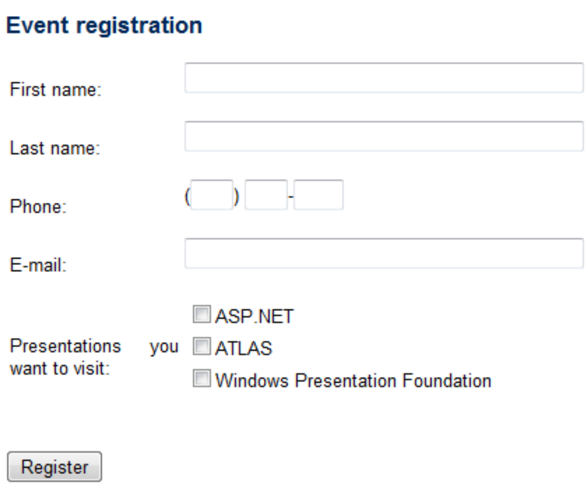
2. In its editing interface, go to the **Form** tab and check the **Use custom form layout** check-box.

3. Click **Generate table layout** button. This will quickly generate a table containing all fields of the form, with appropriate labels, input controls and validation controls. Alternatively, you may use the **Insert label**, **Insert input** and **Insert validation label** buttons on the right to achieve the same result. This gives you the option to include just the fields that you want to appear in the form.

4. With the table generated, you may try altering appearance of the form using actions on the WYSIWYG editor toolbar. For the purpose of this example, add a heading saying *Event registration* above the form, select it and choose **Format: Heading 2** in the WYSIWYG editor toolbar. You may also delete the first line of the table (*EventRegistrationID*), which is not wanted to appear on the public form. The result should look as in the screenshot below.



5. Click **Save** to save the new layout. Now if you add the form to the live site, it should look as in the following screenshot:

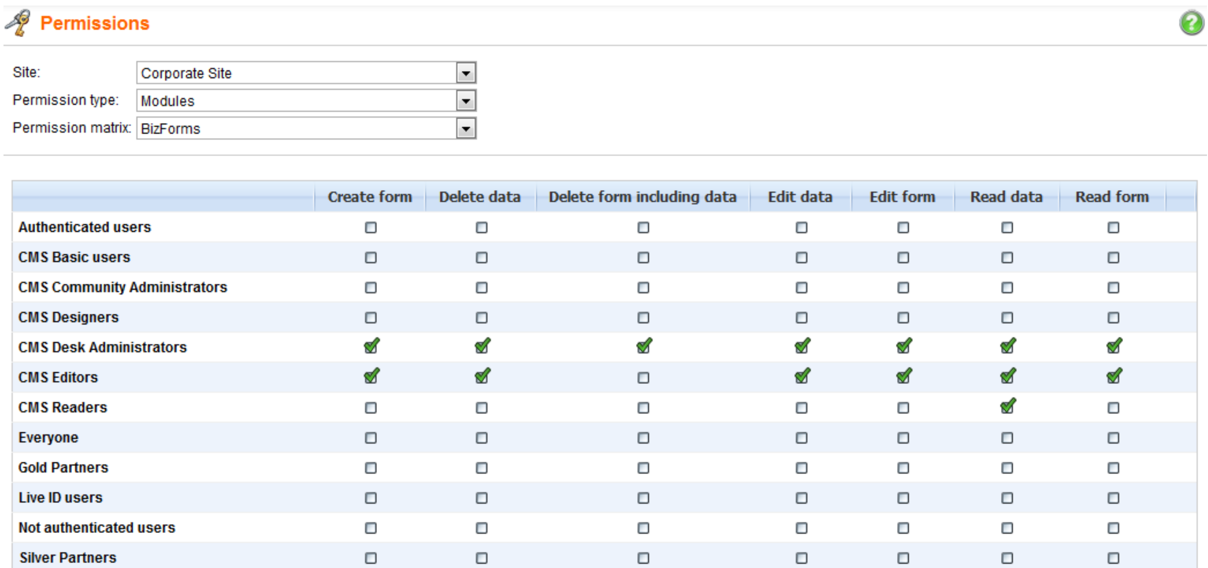


8.8.8 Security

Access to the BizForms module can be managed in **CMS Desk -> Administration -> Permissions**, after you select the **Modules -> BizForms** permission matrix. The BizForms module has the following permissions:

- **Create form** - members of the role are allowed to create new BizForms
- **Delete data** - members of the role are allowed to delete existing BizForm records
- **Delete form including data** - members of the role are allowed to delete BizForms including stored records

- **Edit data** - members of the role are allowed to create and edit BizForm records
- **Edit form** - members of the role are allowed to edit BizForm configuration, fields and layout (not the actual records)
- **Read data** - members of the role are allowed to view BizForm records
- **Read form** - members of the role are allowed to view BizForm configuration, fields and layout (not the actual records)



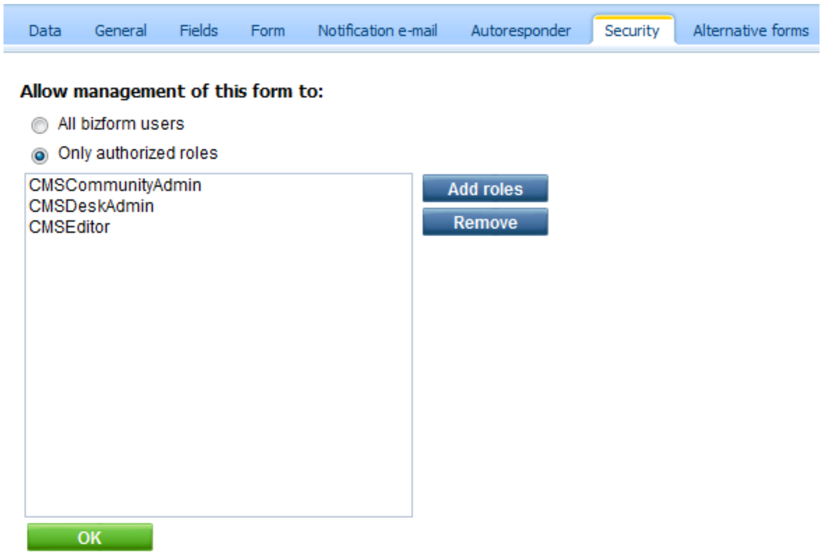
	Create form	Delete data	Delete form including data	Edit data	Edit form	Read data	Read form
Authenticated users	☐	☐	☐	☐	☐	☐	☐
CMS Basic users	☐	☐	☐	☐	☐	☐	☐
CMS Community Administrators	☐	☐	☐	☐	☐	☐	☐
CMS Designers	☐	☐	☐	☐	☐	☐	☐
CMS Desk Administrators	✓	✓	✓	✓	✓	✓	✓
CMS Editors	✓	✓	☐	✓	✓	✓	✓
CMS Readers	☐	☐	☐	☐	☐	✓	☐
Everyone	☐	☐	☐	☐	☐	☐	☐
Gold Partners	☐	☐	☐	☐	☐	☐	☐
Live ID users	☐	☐	☐	☐	☐	☐	☐
Not authenticated users	☐	☐	☐	☐	☐	☐	☐
Silver Partners	☐	☐	☐	☐	☐	☐	☐

Security for particular forms

Roles which are authorized to read and modify a form and its data can also be specified on form level. To do this, edit ( ) a particular form and switch to its **Security** tab. The following two options are available there:

- **All bizform users** - all users with access to the **Tools -> BizForms** section can edit the form
- **Only authenticated users** - only members of roles added to the box will be allowed to edit the form

Please note: General module permissions for the BizForms module (described above) must be granted to the role first. Then, you can further customize access to particular BizForms using the form-level settings. The fact that the role is granted with access to a particular form is not sufficient - the form-level settings only define if the particular form will be listed in **Tools -> BizForms**.



8.8.9 BizForms internals and API

8.8.9.1 Database tables and API classes

The BizForms module uses the following database tables:

- **BizForm_<website code name>_<table name> -** each BizForm has its own database table with name in this format, e.g. *BizForm_CorporateSite_EventRegistration*; this table contains all fields that you specified on the **Fields** tab and you can modify stored data using direct access to the table (there are no dependencies).
- **CMS_Class** - each BizForm has an associated record in this table; you can recognize them by the *BizForm.* prefix in the *ClassName* column

The BizForms API is provided by the following **CMS.SiteProvider** namespace classes:

- **BizFormInfo**, **BizFormInfoProvider -** these classes provide functionality for managing BizForms

The following topics show examples of how these classes can be used:

- Creating a new record
- Updating a record
- Deleting a record
- Customization possibilities

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.8.9.2 Creating a new record

The sample code below demonstrates how a new record can be added to a BizForm using the API.

[C#]

```
using CMS.FormEngine;
using CMS.SettingsProvider;
using CMS.DataEngine;
using CMS.GlobalHelper;

...

        string bizFormName = "TestingSiteContactUs";
        string siteName = "CMSTestingSite";

        // Read BizFrom definition
        BizFormInfo bfi = BizFormInfoProvider.GetBizFormInfo(bizFormName,
siteName);

        if (bfi != null)
        {
            // Read data type definition
            DataClassInfo dci = DataClassInfoProvider.GetDataClass(bfi.
FormClassID);

            if (dci != null)
            {
                GeneralConnection genConn = ConnectionHelper.GetConnection();
                // create a new record in memory (new DataClass object)
                DataClass formRecord = new DataClass(dci.ClassName, genConn);

                // Insert some data
                formRecord.SetValue("FirstName", "Alice");
                formRecord.SetValue("LastName", "Cooper");
                formRecord.SetValue("Email", "alice@email.com");
                formRecord.SetValue("Message", "Hallo world");
                formRecord.SetValue("FormInserted", DateTime.Now);
                formRecord.SetValue("FormUpdated", DateTime.Now);

                // Insert the new record in the database
                formRecord.Insert();

                // Update number of entries in BizFormInfo
                bfi.FormItems++;
                BizFormInfoProvider.SetBizFormInfo(bfi);
            }
        }
```

8.8.9.3 Updating a record

This code example shows how an existing BizForm record can be modified.

[C#]

```
using CMS.FormEngine;
using CMS.SettingsProvider;
using CMS.DataEngine;
using CMS.GlobalHelper;

...

        string bizFormName = "TestingSiteContactUs";
        string siteName = "CMSTestingSite";

        // Read BizForm definition
        BizFormInfo bfi = BizFormInfoProvider.GetBizFormInfo(bizFormName,
siteName);

        if (bfi != null)
        {
            // Read data type definition
            DataClassInfo dci = DataClassInfoProvider.GetDataClass(bfi.
FormClassID);

            if (dci != null)
            {
                // Get all bizform data
                GeneralConnection genConn = ConnectionHelper.GetConnection();
                DataSet ds = genConn.ExecuteQuery(dci.ClassName + ".selectall",
null, null, null);

                if (!DataHelper.DataSourceIsEmpty(ds))
                {
                    // Get ID of the first record
                    int formRecordID = ValidationHelper.GetInteger(ds.Tables[0].
Rows[0][0], 0);
                    // Get the record with ID of the first row record
                    DataClass formRecord = new DataClass(dci.ClassName,
formRecordID, genConn);

                    if (!formRecord.IsEmpty())
                    {
                        // Set new field values
                        formRecord.SetValue("FirstName", "Bob");
                        formRecord.SetValue("LastName", "Marley");
                        formRecord.SetValue("Email", "bob@email.com");
                        formRecord.SetValue("Message", "Good job:)");
                        formRecord.SetValue("FormUpdated", DateTime.Now);

                        // Save updates in the database
                        formRecord.Update();

                        lblInfo.Text = "The first record was updated
successfully.";
                    }
                }
```

```
                else
                {
                    lblInfo.Text = "No data found.";
                }
            }
        }
```

8.8.9.4 Deleting a record

This code example shows how a BizForm record can be deleted.

[C#]

```
using CMS.FormEngine;
using CMS.SettingsProvider;
using CMS.DataEngine;
using CMS.GlobalHelper;

...

        string bizFormName = "TestingSiteContactUs";
        string siteName = "CMSTestingSite";

        // Get BizForm definition
        BizFormInfo bfi = BizFormInfoProvider.GetBizFormInfo(bizFormName,
siteName);

        if (bfi != null)
        {
            // Get data type definition
            DataClassInfo dci = DataClassInfoProvider.GetDataClass(bfi.
FormClassID);
            if (dci != null)
            {
                // Get all bizform data
                GeneralConnection genConn = ConnectionHelper.GetConnection();
                DataSet ds = genConn.ExecuteQuery(dci.ClassName + ".selectall",
null, null, null);

                if (!DataHelper.DataSourceIsEmpty(ds))
                {
                    // Get ID of the first record
                    int formRecordID = ValidationHelper.GetInteger(ds.Tables[0].
Rows[0][0], 0);
                    // Get the record with specified ID
                    DataClass formRecord = new DataClass(dci.ClassName,
formRecordID, genConn);

                    if (!formRecord.IsEmpty())
                    {
                        // Delete the first record.
                        formRecord.Delete();
```

```
                    // Update number of entries in BizFormInfo
                    bfi.FormItems--;
                    BizFormInfoProvider.SetBizFormInfo(bfi);

                    lblInfo.Text = "The record was deleted successfully.";
                }
            }
            else
            {
                lblInfo.Text = "No data found.";
            }
        }
    }
```

8.8.9.5 Customization possibilities

This topic provides information on how BizForms behavior can be modified with your custom code.

Event Handling

You can run custom actions when a BizForm form record is created, updated or deleted. There are two ways how you can handle these events:

1. Handling BizForm control events

In this case, you place the BizForm control on a user control or web part and specify a code name of the required BizForm in its **FormName** property. Then, you can handle the BizForm events, such as **OnAfterSave**, **OnBeforeValidate**, etc.

In these handlers, you can access form values using the *BizForm.BasicForm.DataRow* property and stop further processing (saving) of the form using the *BizForm.StopProcessing* property. Particular fields, labels and validation error messages can be accessed using the *BizForm.BasicForm.FieldControls*, *BizForm.BasicForm.FieldLabels* and *BizForm.BasicForm.FieldErrorLabels* properties.

Example of this type of customization is located in Development -> Web parts -> Modifying web parts -> Modifying code of standard web parts.

2. Handling global data events

You can write a custom data handler as described in the API programming and Kentico CMS internals -> Data handler (CustomDataHandler class) topic. Such a handler will allow you to run custom code whenever a database record is inserted, updated or deleted. You can use the *dataItem.ClassName* property to check if the record being updated is your BizForm code name (you can find it in the **Form code name** field on the General tab of a form's editing () interface).

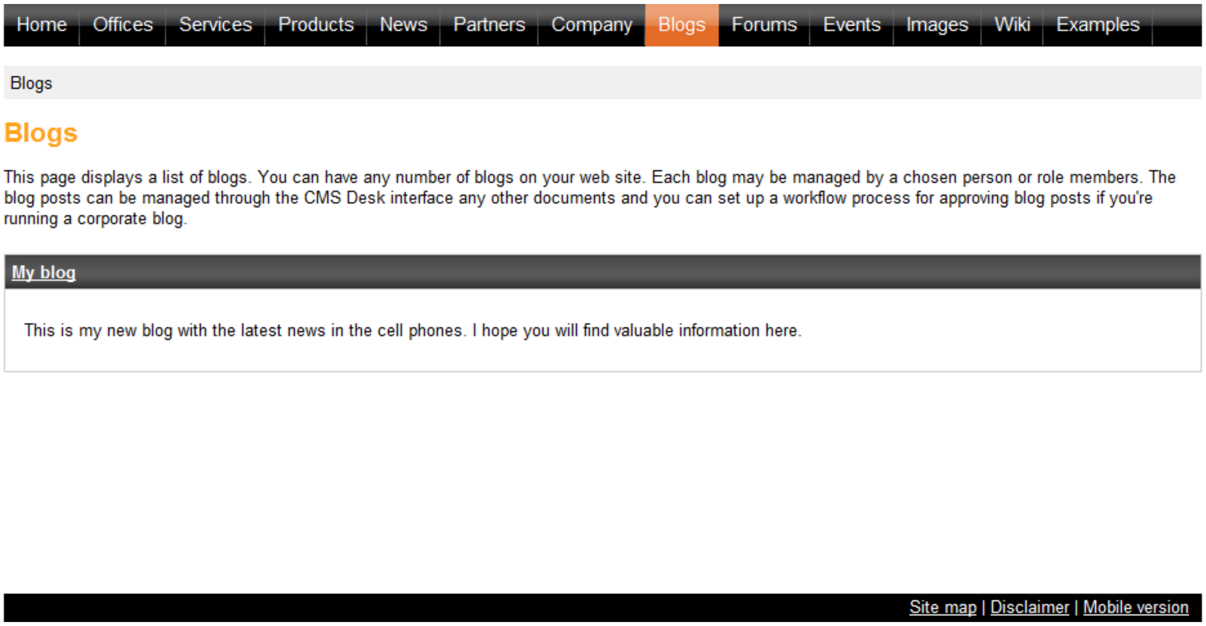
Custom Field Controls

You can create your own field controls as described in chapter Form controls. If you want to make them available in the BizForms field editor, you need to check the box **Show control in BizForms** in the **Form control properties** dialog (in Site Manager -> Development -> Form controls) and choose the default data type.

8.9 Blogs

8.9.1 Overview

The Blogs module allows you to publish a personal or company blog, which is a frequent, chronological publication of one's thoughts and web links. You can publish multiple blogs on the same site and there can be multiple editors for each blog.



The Blogs module fully leverages the standard content management engine, so every blog post you create is a standard document that can be displayed on the website, searched, etc. You can also configure permissions and workflow for every blog as you're used to do with other content.

If you would like to see an example of blog, please refer to a sample blog on the Corporate Site sample website.

- To learn how to create a new blog on the sample Corporate Site, please refer to the Adding a blog to your site topic.
- To learn how to add posts to your blog, please refer to the Adding posts to your blog topic.
- To learn how comments can be moderated, please follow the Moderating comments topic.
- To learn how to configure the layout and design of your blog, please follow the Blog layout and design topic.
- To learn how to enable users to perform selected tasks using the User contributions module web parts, please refer to the On-site management via User contributions topic.
- If you would like to adjust the settings and security of the Blogs module, please refer to the Settings and Security topics respectively.
- To learn how to deal with blog post documents using standard API for documents, please refer to the Managing blogs using Kentico CMS API topic.

Subchapters on trackbacking and on a programming interface enabling co-operation of the Blogs module with external programs, MetaWeblog API, and topics related to Blog comments notifications (Who can be notified, User subscriptions and E-mail templates) are also available in the Blogs chapter.

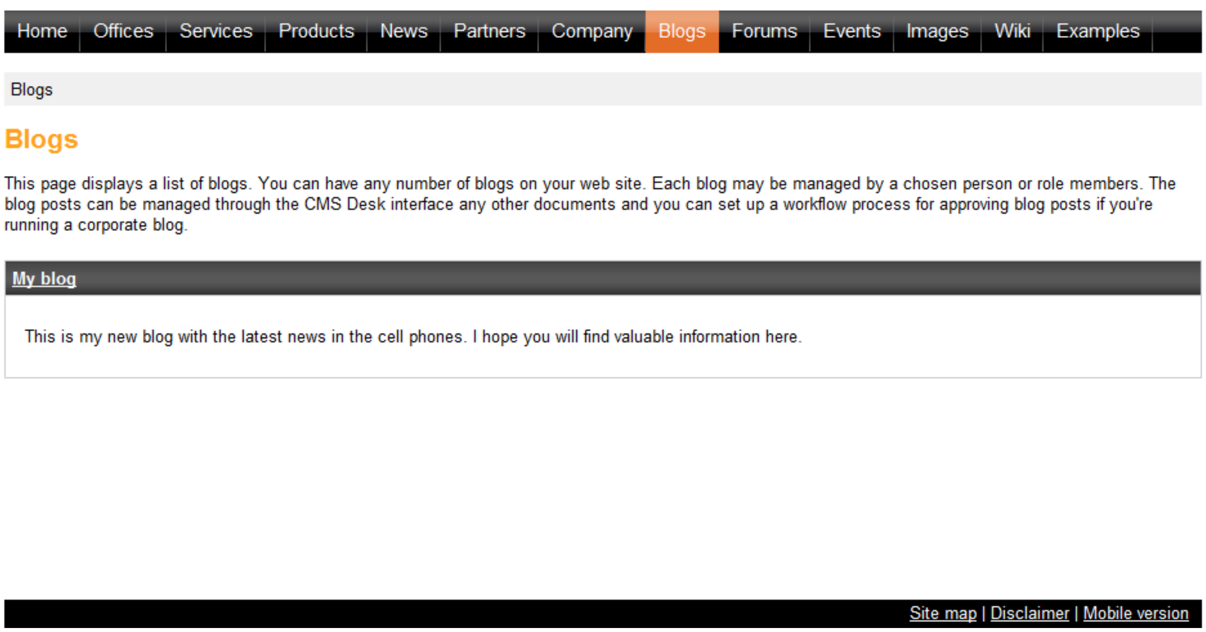
If you would like to compare the Corporate Site sample blog with a more detailed example of blogging, please switch to the Community Site sample website where you can find not only blogs with blog posts but also a tag cloud. The Tag cloud web part is used to display key words, called tags that are associated with a document, and is convenient for simple marking of documents according to various criteria, e.g. your interests.

Kentico CMS Community Site Guide contains some additional examples and tutorials related to blogs:

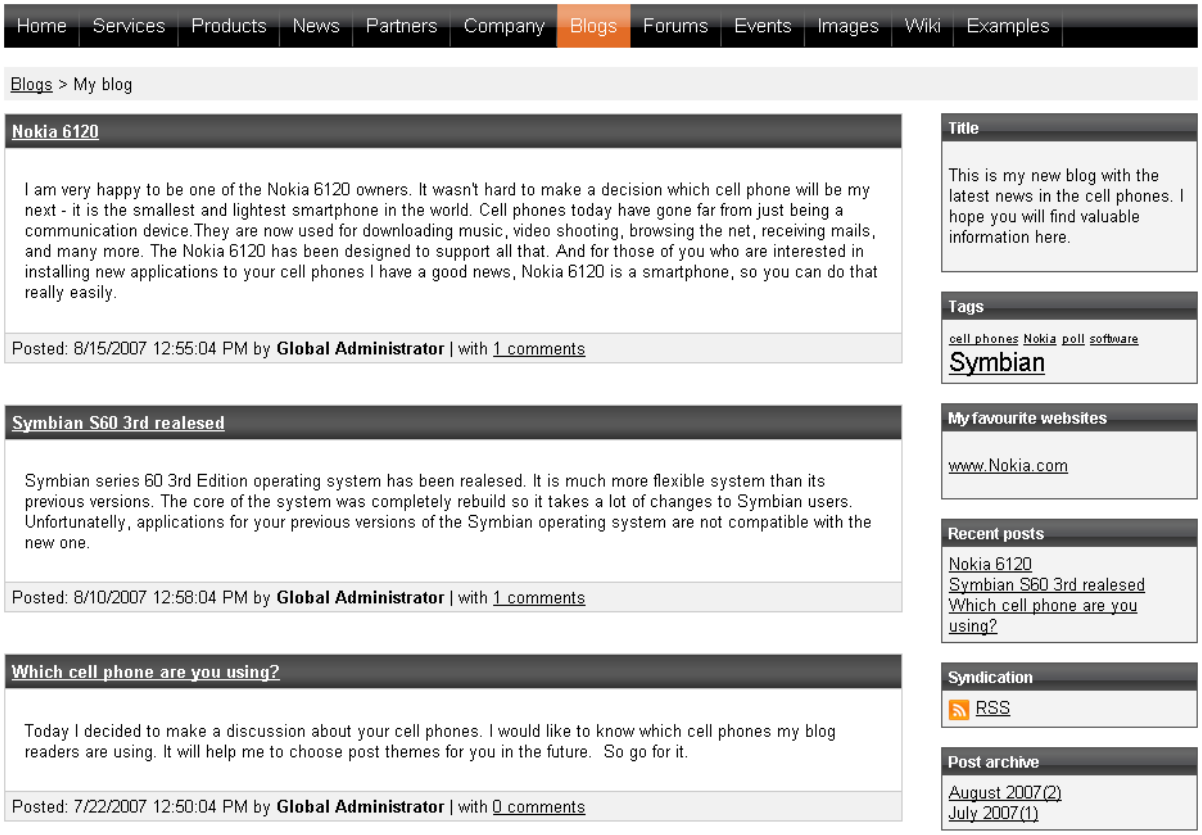
You will learn not only how to create a page on blogs (learn here how to do it) but also how to create a special page where users are redirected when they want to create a new blog (as explained here), a page that displays a list of all blogs on the site (as explained here) or how to create a page where only blog posts corresponding to the selected tag from the tag cloud are displayed (as explained here).

8.9.2 Sample blog

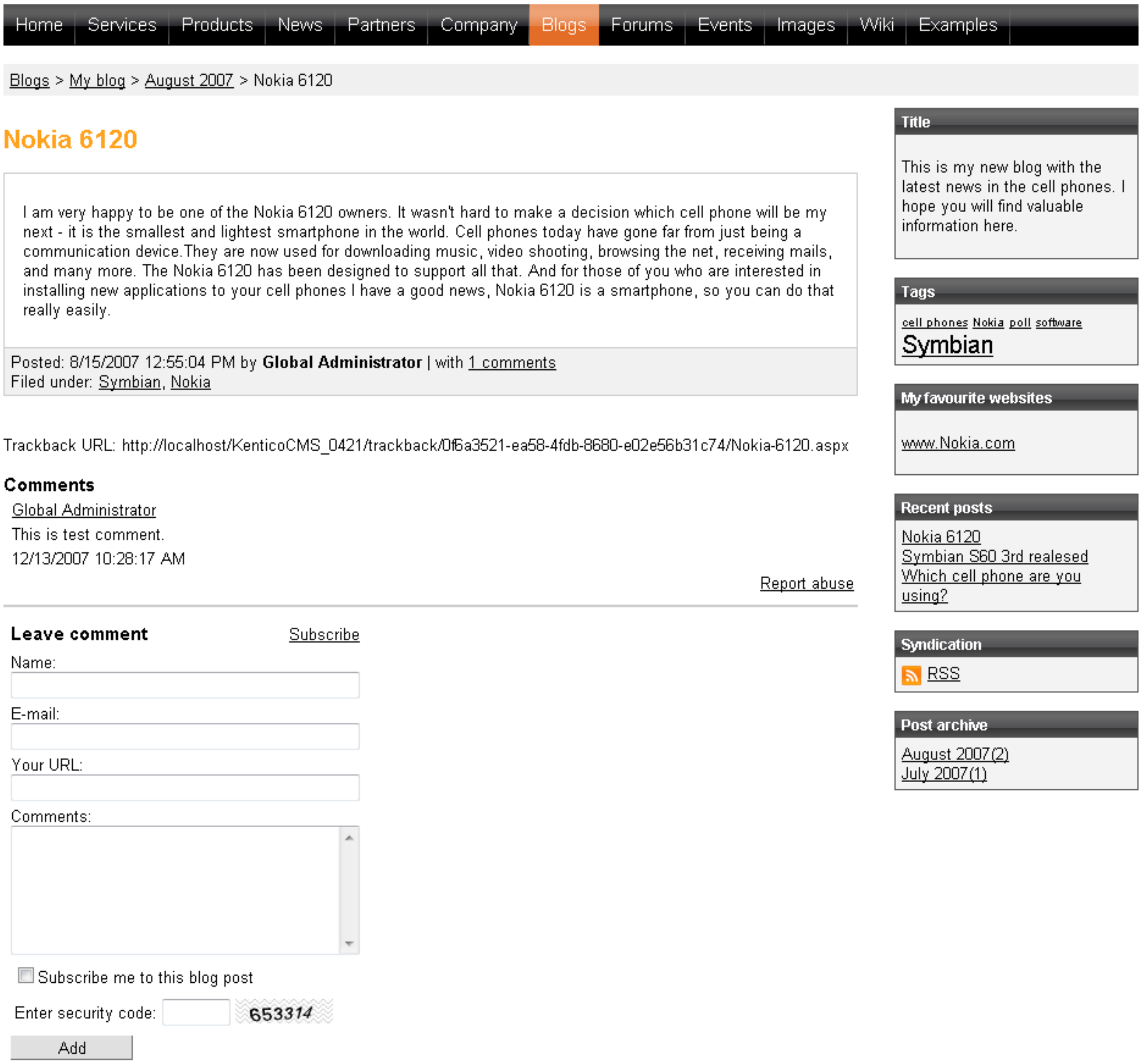
To see a sample blog, on the sample Corporate Site switch to the live site and from the **Main menu** choose **Blogs**. The Blogs page displays a list of blogs, each of which can be accessed by clicking on the particular blog link.



After clicking the chosen blog link, a new page with a list of blog posts and some additional information is displayed. As you can see, the additional information such as blog description, tags, favorite websites, recent posts, RSS feed link and post archive is displayed on the right of the page.



Finally, when you click on some particular blog post link, you can see the text plus the **Comments** section that allows you to read and add comments to the blog post. Additional information related to the current blog is also available.



8.9.3 Adding a blog to your site

In this example, you will learn how to create a new blog in the sample Corporate Site, under the News section.

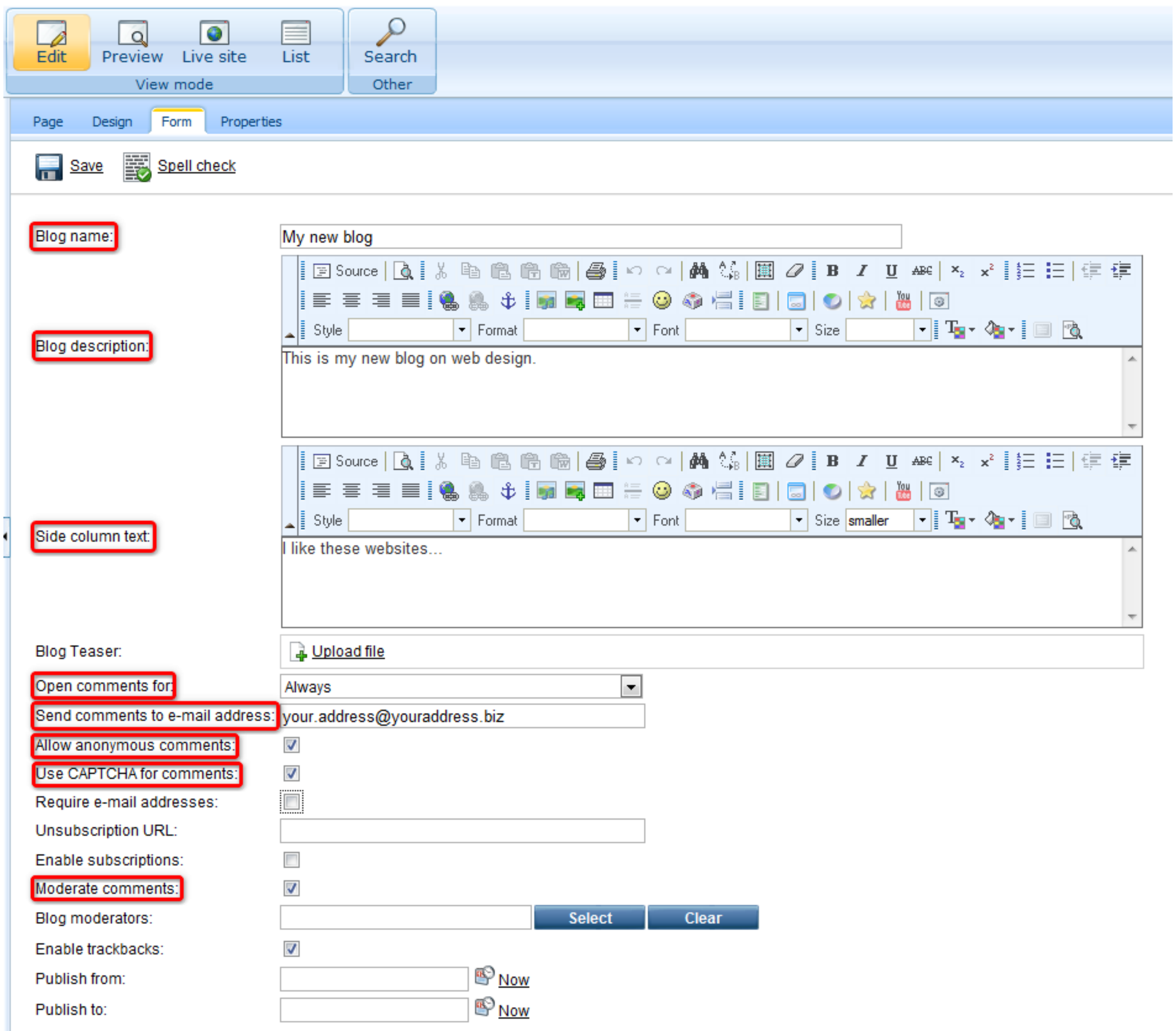
1. Sign in to **CMS Desk**. In the **Content** section, click **/News** and click **New**. Choose to create a new **Blog**. Enter the following details:

- **Blog name:** My new blog
- **Blog description:** This is my new blog on web design.
- **Side column text:** I like these websites…
- **Open comments for:** Always
- **Send comments to e-mail:** *enter your e-mail address*
- **Allow anonymous comments:** yes
- **Use CAPTCHA for comments:** yes (this will require a verification of the number displayed in the picture to avoid spam posts)

- **Moderate comments:** yes

The following fields are also available, but you don't need to enter them for the purpose of this example:

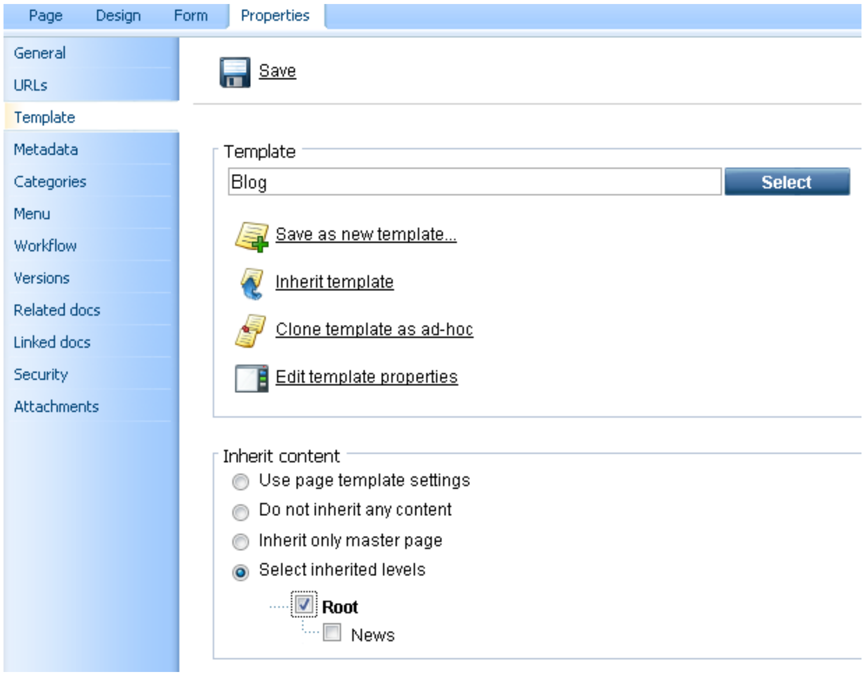
- **Blog teaser:** blog teaser image displayed in blog lists
- **Require e-mail address:** indicates if e-mail address is required when adding a blog comment
- **Unsubscription URL:** URL of a page with the Blog post unsubscripiton web part, which handles blog post unsubscription requests from links contained in blog post notifications
- **Enable subscriptions:** indicates if subscriptions to notifications about new blog comments are enabled
- **Blog moderators:** users who can moderate blog post comments
- **Enable trackbacks:** indicates if the trackbacking feature is enabled
- **Publish from:** date and time from when the blog will be published on the site
- **Publish to:** date and time until when the blog will be published on the site



Click **Save**. You have just created your blog. However, there are two additional steps you need to do:

2. Go to the **Properties -> Template** dialog, click **Select** and choose the **Blog** page template. Set the

Inherit content value to **Select inherited levels** and choose only the **Root**. It ensures that the page displays the master page and then your blog content.



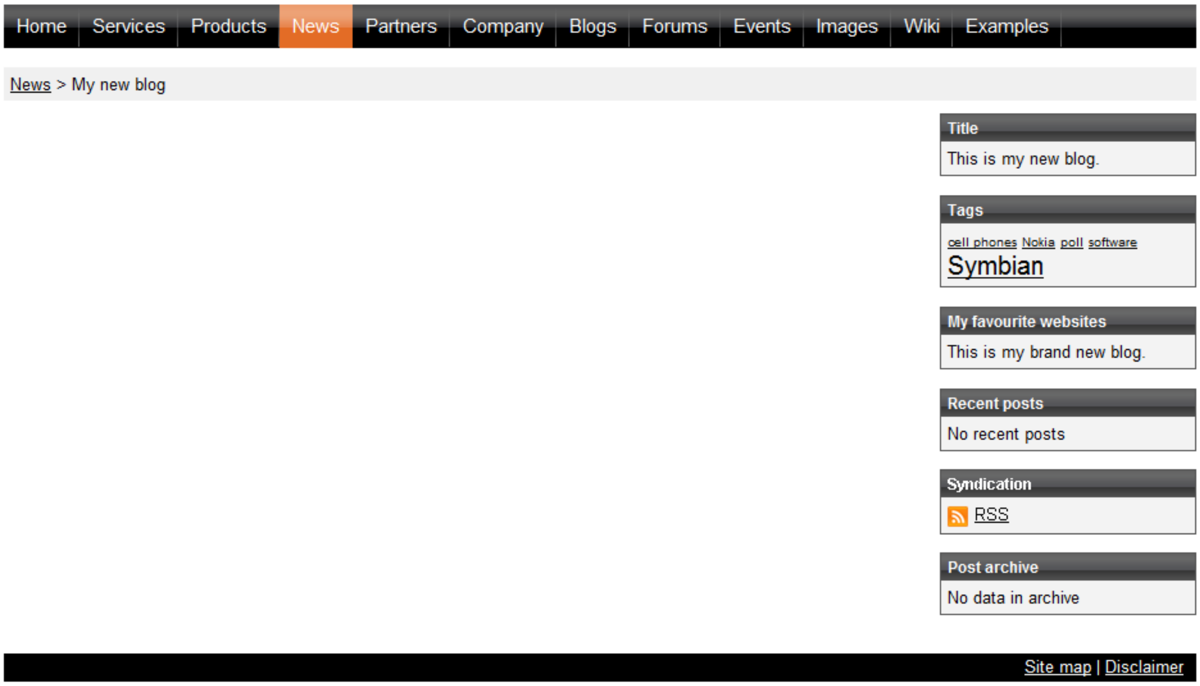
Click **OK**.

3. Choose the root in the content tree and switch to the **Design** tab. Configure the **cmsmenu** web part and set the **Document types** value to this:

cms.menuitem;cms.blog

Click **OK**. It ensures that the main menu contains a link to your blog.

Sign out and choose the **My new blog** item in the main menu (under the **News** section). You will see an empty blog like this:



Adding a blog on the live site

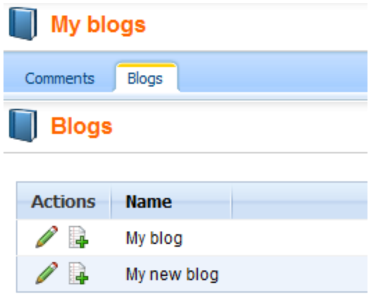
You can enable users to create blogs directly on the live site by adding the **New blog** web part to your site. Like this, users can only add the name and description of their new blog. All other blog properties, listed earlier in this article, will be taken from the web part's properties.

8.9.4 Adding posts to your blog

You can add new posts to your blog in two ways:

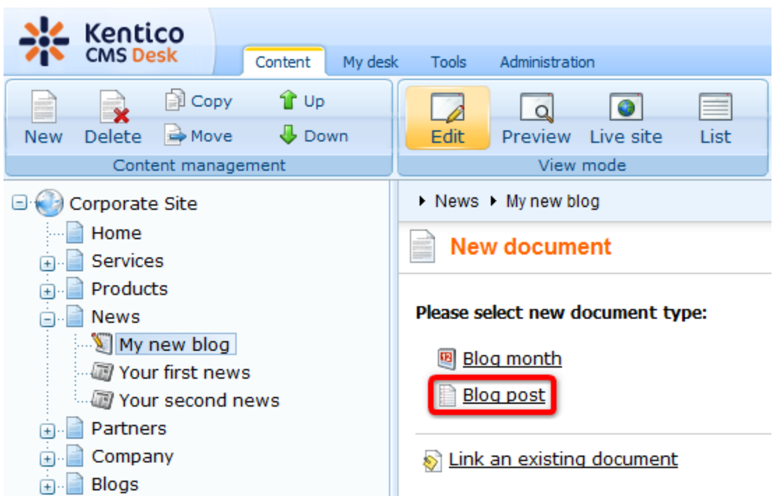
1. From the CMS Desk -> My Desk -> My blogs dialog

This dialog displays the list of blogs you are owner of (you can set the owner in the **Properties -> General** dialog of the blog document). Here you can click the **New post** ( ) icon and you will be displayed with form for inserting the post:



2. In the Content section

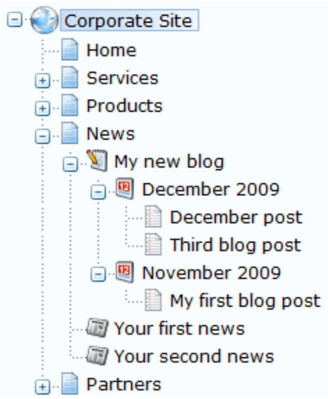
Choose the **Content** tab in **CMS Desk**, select the blog's main document and then click **New**. Choose to create a new **Blog post**:



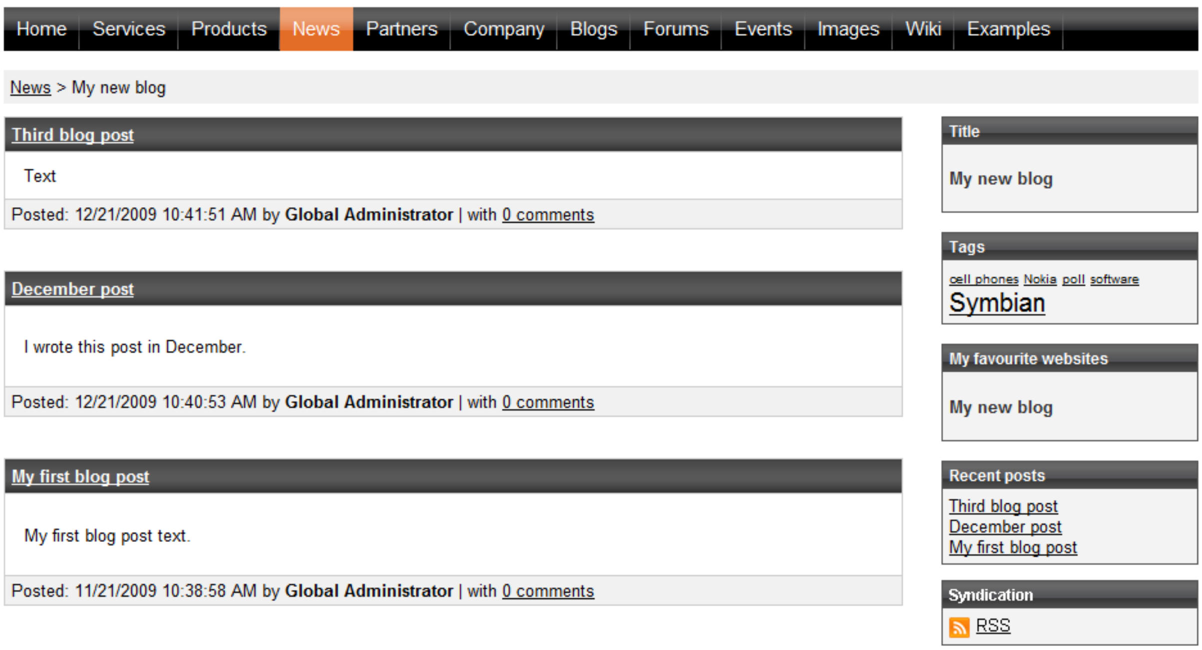
Enter the post content and click **Save**.

Document structure

In both cases, the blog post will be automatically saved under the appropriate month. If the month is not created yet, it's created automatically, so you do not need to care about creating a month. The months allows you to easily organize the blog posts. The structure of **Blog**, **Blog month** and **Blog post** documents looks like this:



When you sign out and go to your blog, you will see a page like this:



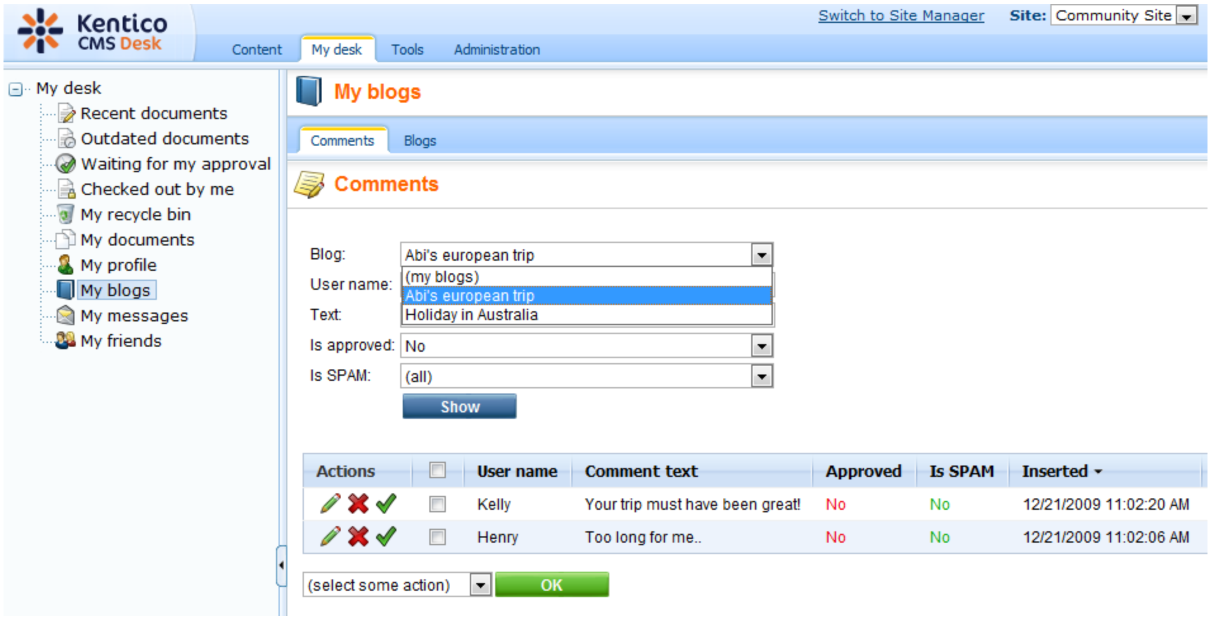
8.9.5 Moderating comments

The comments can be moderated in tree ways:

1. In the My Desk -> My blogs section

In this section, currently logged-in users can manage comments at all their own blogs. Neither the **Read** nor **Manage comments** permissions for the Blogs module are necessary for the user to perform blog comments management in this section.

Comments can be approved by clicking the **Approve** (✔) icon. More comments can be approved at once when you select them by the check-boxes, choose the **Approve** (✔) action in the **Selected items** drop-down list and click **OK**.

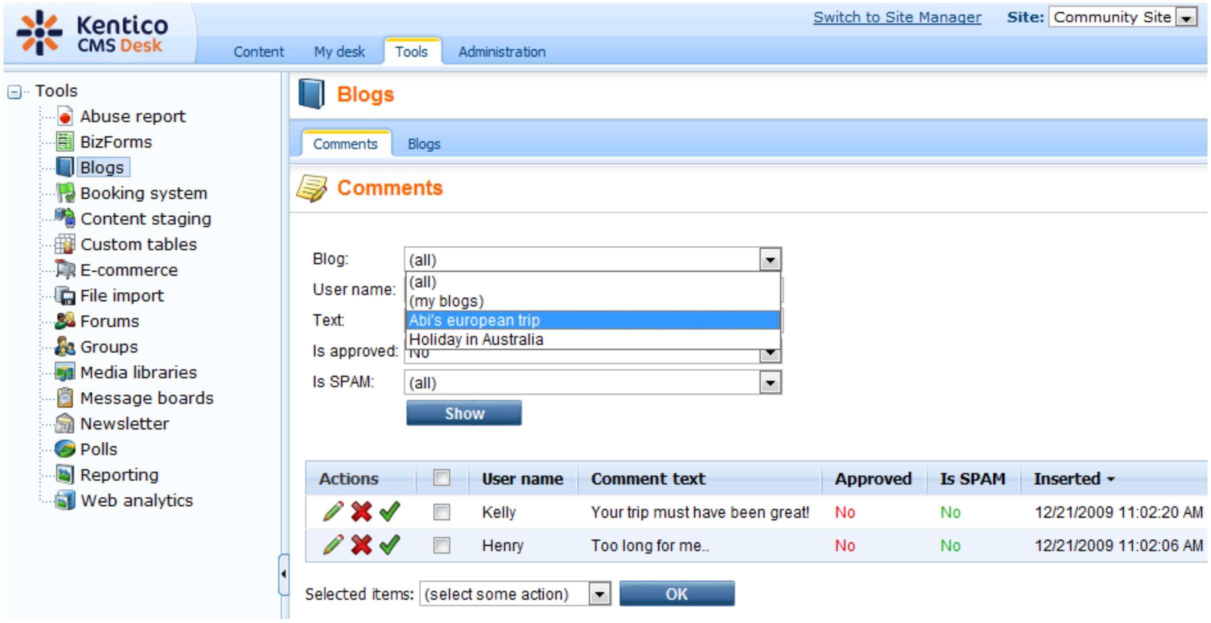


2. In the CMS Desk -> Tools -> Blogs section

In this section, management of blog comments on the site is possible based on the permissions:

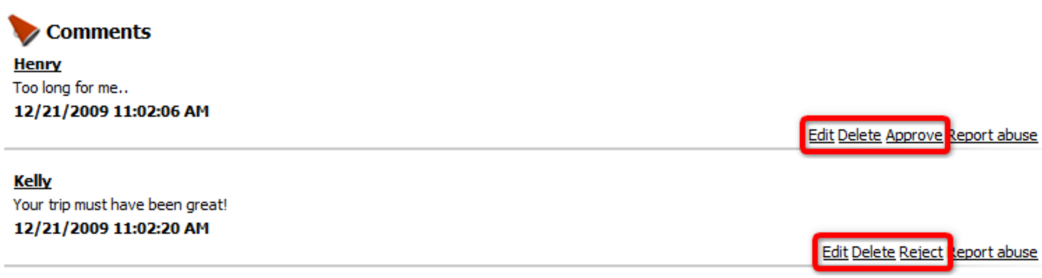
Global administrators and users with the **Manage comments** permission for the Blogs module can manage comments of all blogs on the current site. Blog **moderators** and blog **owners** can manage comments of their own blogs and blogs for that they are moderators. These permissions are reflected by the **Blog** drop-down list, which displays only those blogs where comments can be managed by the current user.

Comments can be approved by clicking the **Approve** (✔) icon. More comments can be approved at once when you select them by the check-boxes, choose the **Approve** (✔) action in the **Selected items** drop-down list and click **OK**.



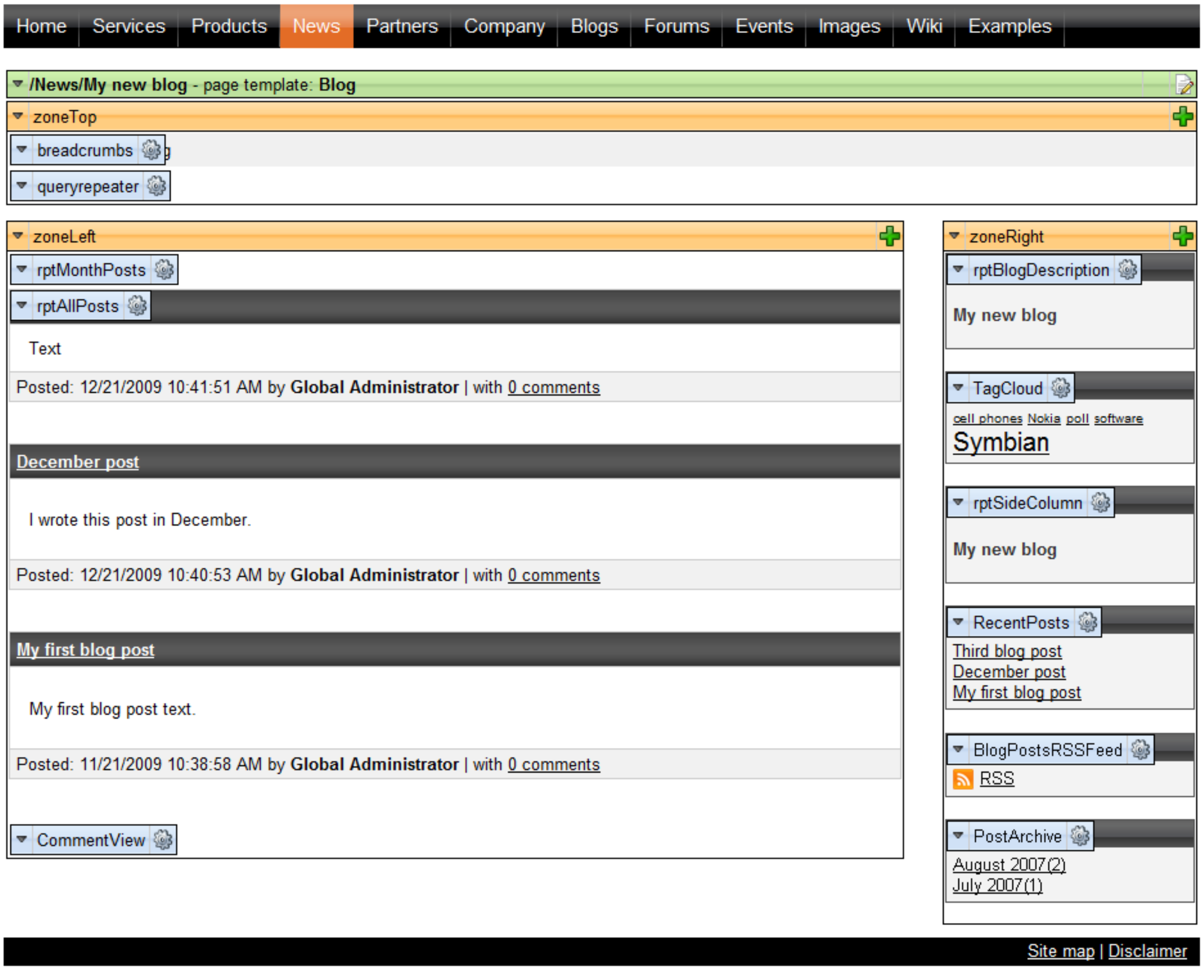
3. On-site management

When you're signed in and are entitled to manage the comments for the given blog, you are displayed with **Edit**, **Delete**, **Approve** and **Reject** buttons. These buttons are displayed to **blog owners**, **moderators** of the current blog, **global administrators** and users with the **Manage comments** permissions for the Blogs module.



8.9.6 Blog layout and design

You can configure the blog layout and design by changing the page template of the blog document. Locate your blog in the **CMS Desk -> Content** section and choose the **Design** tab. You will see a page like this:



The blog is displayed using the Blog page template. The web parts ensure displaying of the blog content, comments and boxes in the right-hand column.

There are two repeaters to display the posts:

- **rptMonthPosts** - this repeater is used to display posts in the given month
- **rptAllPosts** - this repeater is used to display the latest N posts when the blog is displayed without selecting a particular month

If you need to modify the layout of the blog posts, you need to modify the transformations. By default, the following transformations are used:

- **cms.blog.PostPreview** - the view in the list of posts
- **cms.blogpost.Default** - the detailed view

8.9.7 On-site management via User contributions

You can enable users to perform the following tasks using the User contributions module web parts:

- **Edit blog properties**
- **Edit or delete blog posts**

- **Add new blog posts**

A live example of this can be found on the sample **Community Starter site**, on the **Blogs** title page. If you go to CMS Desk's **Design** tab and view the page, you will see the following web parts, which enable the above mentioned functionalities:

- **EditBlog** - Edit contribution web part set up for enabling blog properties editing
- **EditBlogPost** - Edit contribution web part set up for enabling blog post editing
- **NewBlogPost** - Contribution list web part set up for enabling adding of blog posts

These web parts are inherited by the underlaying pages, which means they will be displayed when a blog or blog post is displayed. The following sections explain how to enable users to perform these tasks on the live site.

Edit blog properties

1. Create an **alternative form** for the **Blog** document type, which will contain the fields that you want to let users modify. If you are not familiar with the Alternative forms concept, please refer to the Alternative forms chapter first.

2. Add the **Edit contribution** web part to the blogs section title page and set its following properties:

- **Show for document types** - CMS.Blog
- **Alternative form name** - code name of the alternative form created in step 1
- **Edit button label** - Edit blog
- **Allow editing by users** - Document owner; this is the setting that makes the most sense as blog properties should be edited only by the blog's owner

If you go to the live site and display some blog placed under the blogs section title page, you should see the **Edit blog** link as highlighted in the screenshot below. After clicking the link, the alternative form will be displayed, letting blog owners edit the properties of the blog.

Add new blog posts

1. Create an **alternative form** for the **Blog post** document type, which will contain the fields that you want to let users to specify when creating the blog post. If you are not familiar with the Alternative forms concept, please refer to the Alternative forms chapter first.

2. Add the **Contributions list** web part to the blogs section title page. You only need to set the following properties, as the web part will not be used for displaying blog posts, but only for their inserting.

- **Show for document types** - CMS.Blog
- **Allowed new document types** - CMS.BlogPost
- **Alternative form name** - code name of the alternative form created in step 1
- **New item button label** - New blog post
- **Allow insert** - enable
- **Allow editing by users** - Document owner; this is the setting that makes the most sense as blog posts should be edited only by the blog's owner

If you go to the live site and display some blog placed under the blogs section title page, you should see the **New blog post** link as highlighted in the screenshot below. After clicking the link, the alternative form will be displayed, letting blog owners add new blog posts directly from the live site.

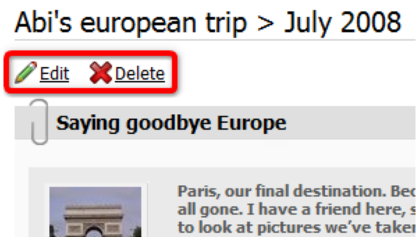
Edit or delete blog posts

1. Create an **alternative form** for the **Blog post** document type, which will contain the fields that you want to let users modify. If you are not familiar with the Alternative forms concept, please refer to the Alternative forms chapter first.

2. Add the **Edit contribution** web part to the blogs section title page and set its following properties:

- **Show for document types** - CMS.BlogPost
- **Alternative form name** - code name of the alternative form created in step 1
- **Edit button label** - Edit
- **Allow delete** - enable
- **Allow editing by users** - Document owner; this is the setting that makes the most sense as blog posts should be edited only by the blog's owner

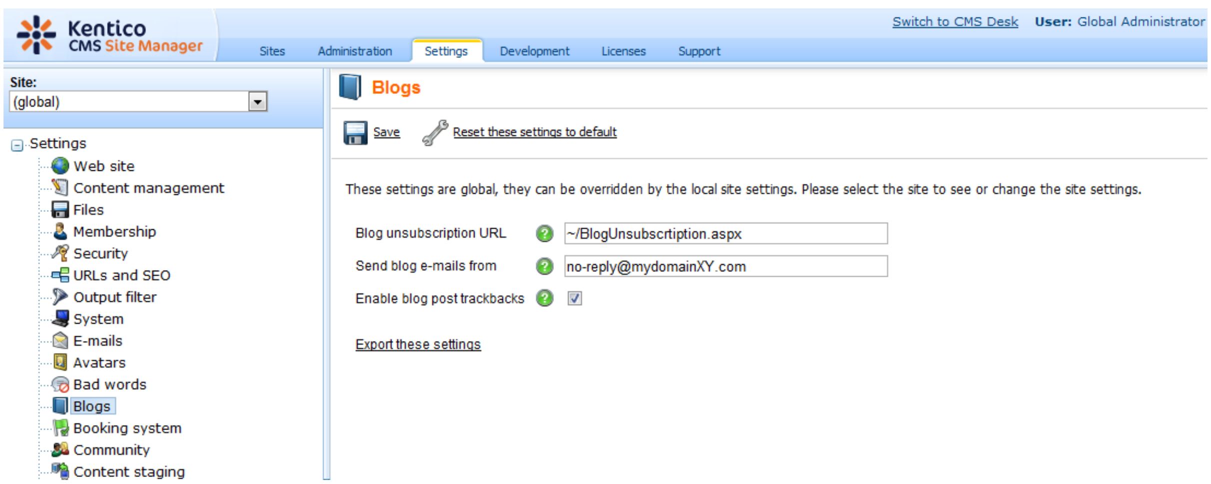
If you go to the live site and display some blog post placed under the blogs section title page, you should see the **Edit** and **Delete** links as highlighted in the screenshot below. After clicking the **Edit** link, the alternative form will be displayed, letting blog owners edit the blog post. Using the **Delete** link, they can delete the blog post.



8.9.8 Settings

Settings of the Blogs module can be done in **Site Manager -> Settings -> Blogs**. The following settings are available:

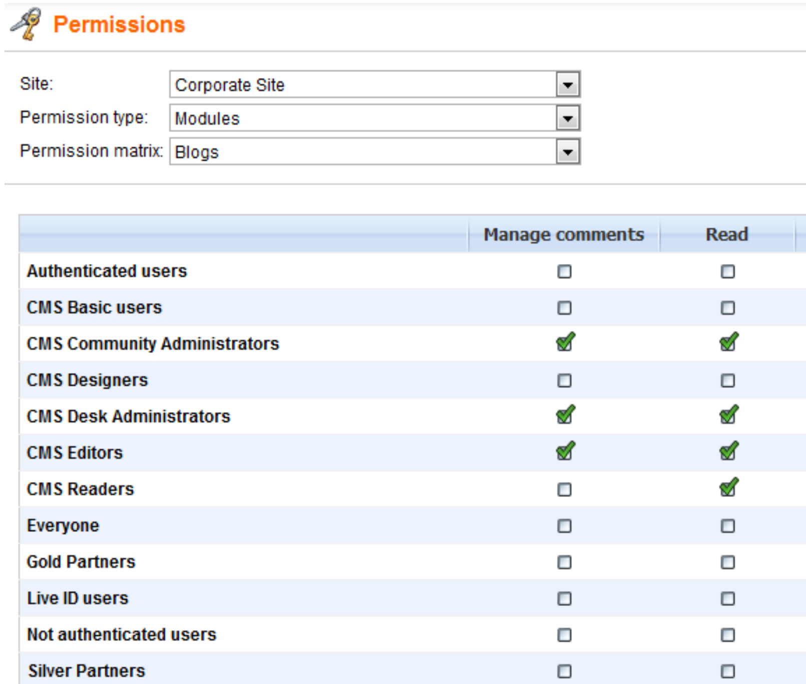
- **Blog unsubscription URL** - URL of a page on which the **Blog post unsubscription** web part is placed; this is a special web part used for handling unsubscriptions from receiving notifications about new blog posts
- **Send blog e-mails from** - e-mail address that will be used as the Sender ('From') address of notification e-mails
- **Enable blog post trackbacks** - If checked, specified blog posts are pinged after the new blog post is saved; turn off this setting if you are creating your site on the production server to avoid creating trackbacks to your production server



8.9.9 Security

The Blogs module leverages the standard security used for all documents, as blogs are actually documents in the content tree. The Blogs module also has the following permissions in **Site Manager -> Administration -> Permissions**:

- **Manage comments** - allows members of the role to manage blog post comments
- **Read** - allows members of the role to view blog posts and blog post comments via the administration interface


Permissions

Site: Corporate Site

Permission type: Modules

Permission matrix: Blogs

	Manage comments	Read
Authenticated users	☐	☐
CMS Basic users	☐	☐
CMS Community Administrators	✓	✓
CMS Designers	☐	☐
CMS Desk Administrators	✓	✓
CMS Editors	✓	✓
CMS Readers	☐	✓
Everyone	☐	☐
Gold Partners	☐	☐
Live ID users	☐	☐
Not authenticated users	☐	☐
Silver Partners	☐	☐

To manage blog posts, you need to be one of the following:

- global administrator
- owner of the blog
- blog moderator
- granted with the **Manage comments** permission for the Blogs module in the **Administration -> Permissions** dialog

The blogs in the **My Desk -> My blogs** dialog are displayed only to the blog document owner.

8.9.10 Managing blogs using Kentico CMS API

Since the blog posts are standard Kentico CMS documents, you can create, modify, read and delete them using standard API for documents. You can find more details and examples in chapter Content management internals and the subsequent chapters.

The following examples show how you can work with comments using the API:

Adding a new comment

[C#]

```
using CMS.CMSHelper;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.Blogs;
```

```
...

            // Use a valid alias path of a testing post
            string postAlias = "/TestingBlog/October-2007/TestingPost";

            // Prepare the TreeProvider (it must be initialized with user
information when editing
            // document structure)
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);

            CMS.TreeEngine.TreeNode postNode = tree.SelectSingleNode(CMSContext.
CurrentSiteName,
               postAlias, TreeProvider.ALL_CULTURES, true, null, false);
            if (postNode != null)
            {
               // Add new comment to the post
               BlogCommentInfo commentObj = new BlogCommentInfo();
               commentObj.CommentDate = DateTime.Now;
               commentObj.CommentPostDocumentID = postNode.DocumentID;
               commentObj.CommentText = "Some comment text";
               commentObj.CommentUserID = ui.UserID;
               commentObj.CommentUserName = (ui.UserNickName != "") ? ui.
UserNickName : ui.FullName;

              // Save comment
              BlogCommentInfoProvider.SetBlogCommentInfo(commentObj);
              }
```

Editing an existing comment

[C#]

```
using CMS.CMSHelper;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.Blogs;

...

            // Use a valid alias path of a testing post
            string postAlias = "/TestingBlog/October-2007/TestingPost";

            // Prepare the TreeProvider (it must be initialized with user
information when editing
            // document structure)
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);
            DataSet ds = null;

            // Get the parent post document
```

```
            CMS.TreeEngine.TreeNode postNode = tree.SelectSingleNode(CMSContext.
CurrentSiteName, postAlias, TreeProvider.ALL_CULTURES, true, null, false);
            if (postNode != null)
            {
                // Get all post comments
                ds = BlogCommentInfoProvider.GetPostComments(postNode.DocumentID,
false);
                if (!DataHelper.DataSourceIsEmpty(ds))
                {
                    // Set all post comments as approved
                    foreach (DataRow dr in ds.Tables[0].Rows)
                    {
                        // Edit comment and save it
                        BlogCommentInfo commentObj = new BlogCommentInfo(dr);
                        commentObj.CommentApprovedByUserID = ui.UserID;
                        BlogCommentInfoProvider.SetBlogCommentInfo(commentObj);
                    }
                }
            }
```

Deleting a comment

[C#]

```
using CMS.CMSHelper;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.Blogs;

...

        // Use a valid alias path of a testing post
        string postAlias = "/TestingBlog/October-2007/TestingPost";

        // Prepare the TreeProvider (it must be initialized with user information
when editing
        // document structure)
        UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
        TreeProvider tree = new TreeProvider(ui);
        DataSet ds = null;


        // Get the parent post document
        CMS.TreeEngine.TreeNode postNode = tree.SelectSingleNode(CMSContext.
CurrentSiteName,
            postAlias, TreeProvider.ALL_CULTURES, true, null, false);
        if (postNode != null)
        {
            // Get all post comments
            ds = BlogCommentInfoProvider.GetPostComments(postNode.DocumentID,
false);
```

```
            if (!DataHelper.DataSourceIsEmpty(ds))
            {
                // Delete all post comments
                foreach (DataRow dr in ds.Tables[0].Rows)
                {
                    // Delete comment
                    BlogCommentInfo commentObj = new BlogCommentInfo(dr);
                    BlogCommentInfoProvider.DeleteBlogCommentInfo(commentObj.
CommentID);
                }
            }
        }
```

8.9.11 Trackbacks

8.9.11.1 Trackbacks overview

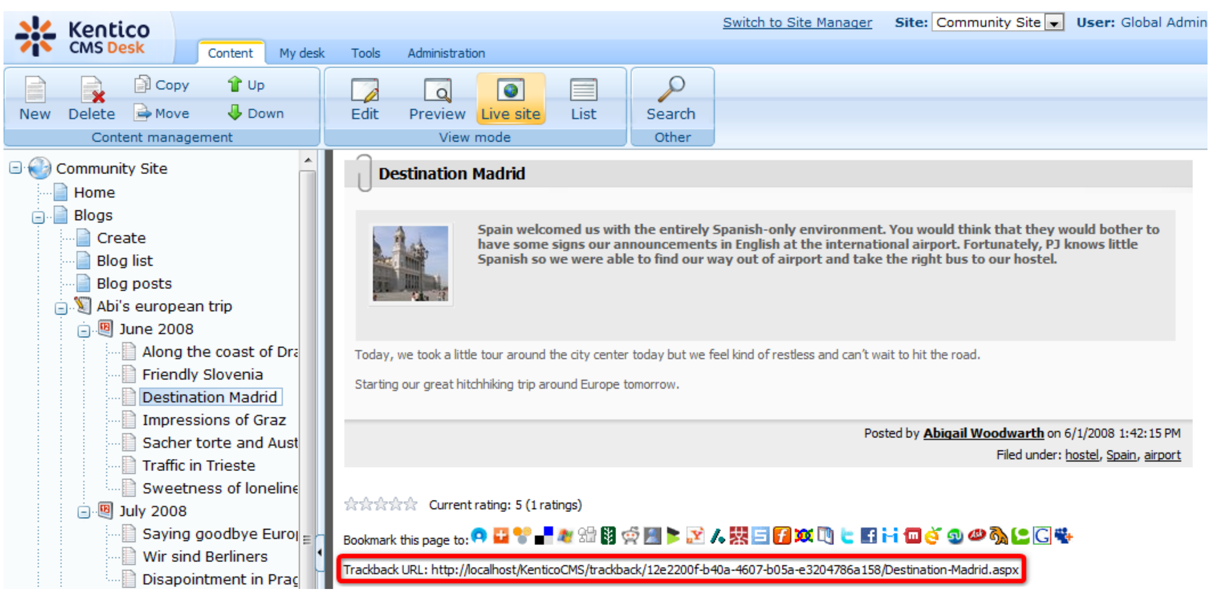
The **Trackbacks** feature allows sending of blog post links to other blogs when a new blog post is created. The link is typically, but not always, added as a blog comment in the target blog, as implemented in Kentico CMS. This can be performed not only within the same site, but also to completely different sites created not only with Kentico CMS.

Trackbacks are typically used to let readers of a topic-related blog posts know about your new blog post. You can find detailed information on trackbacks on the following Wikipedia page: http://en.wikipedia.org/wiki/Trackback

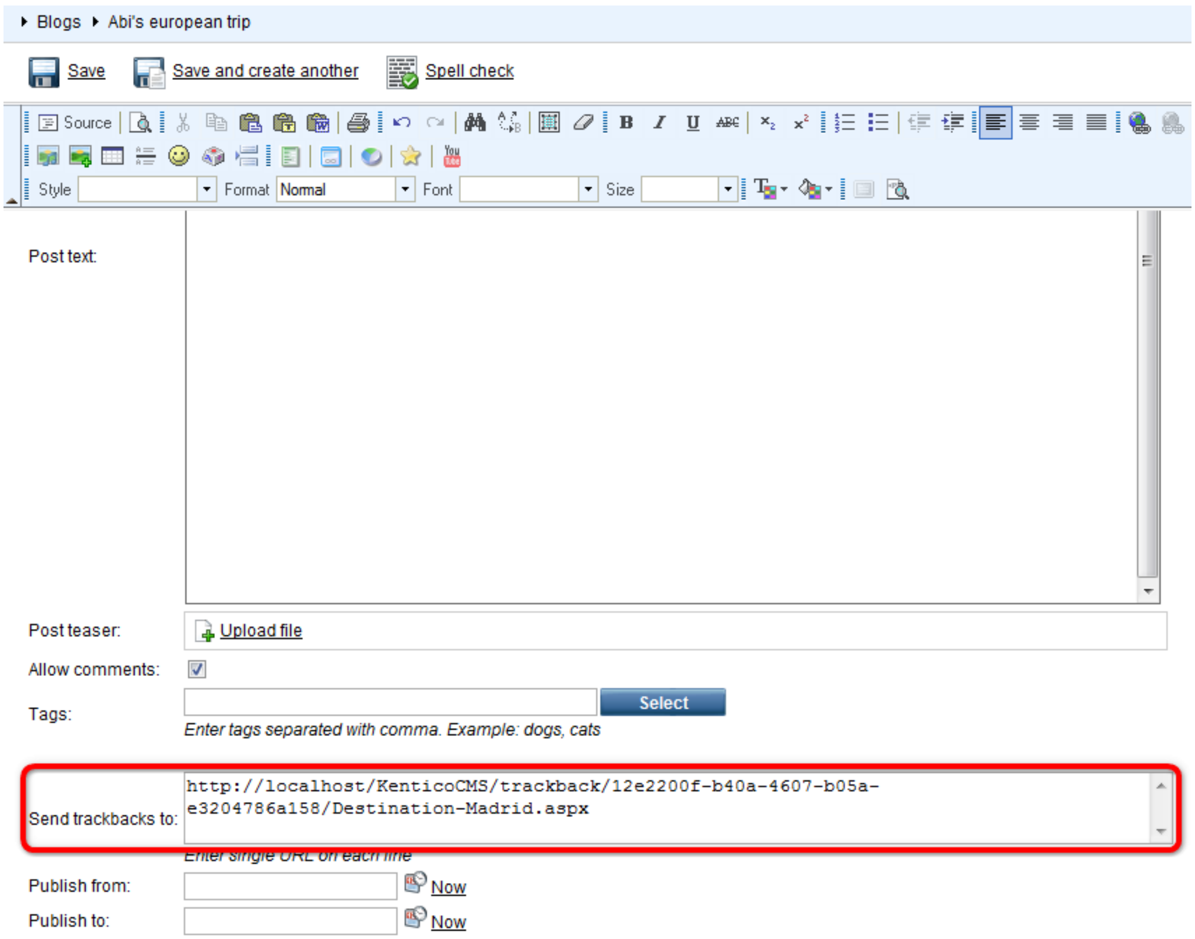
Read here to learn how to enable trackbacks on your site.

How it works

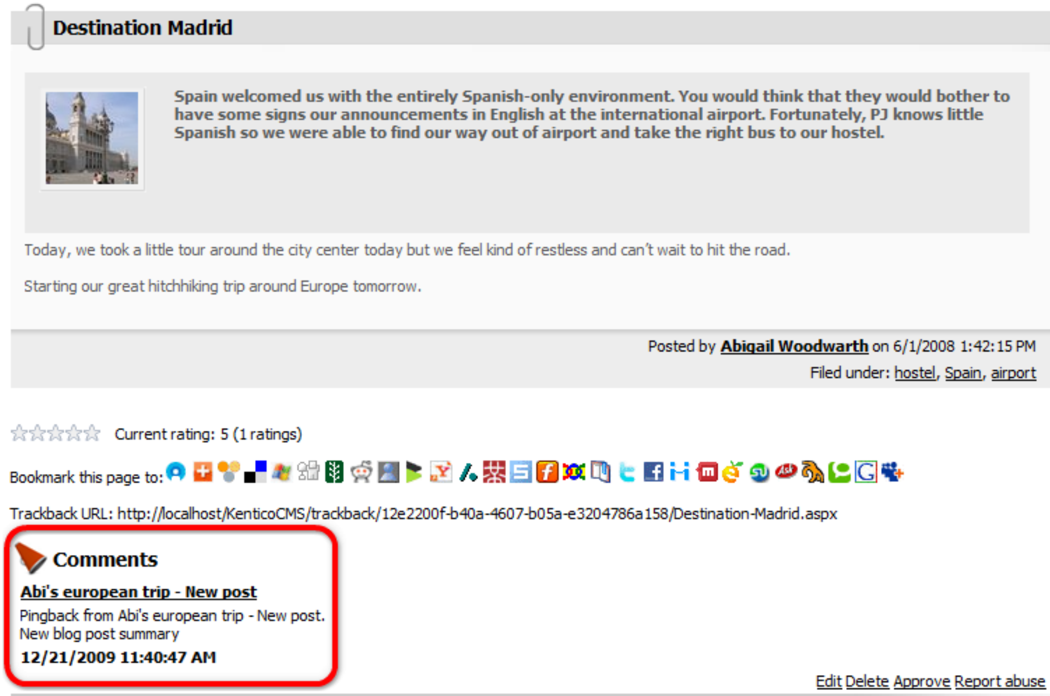
1. There is a Trackback URL published below the blog post. These URLs can be found on various blogs at sites developed not only with Kentico CMS.



2. When users are creating a new blog post, they can add such URLs into the **Send trackbacks to** field. More than one trackback URL can be entered, each on a new line.



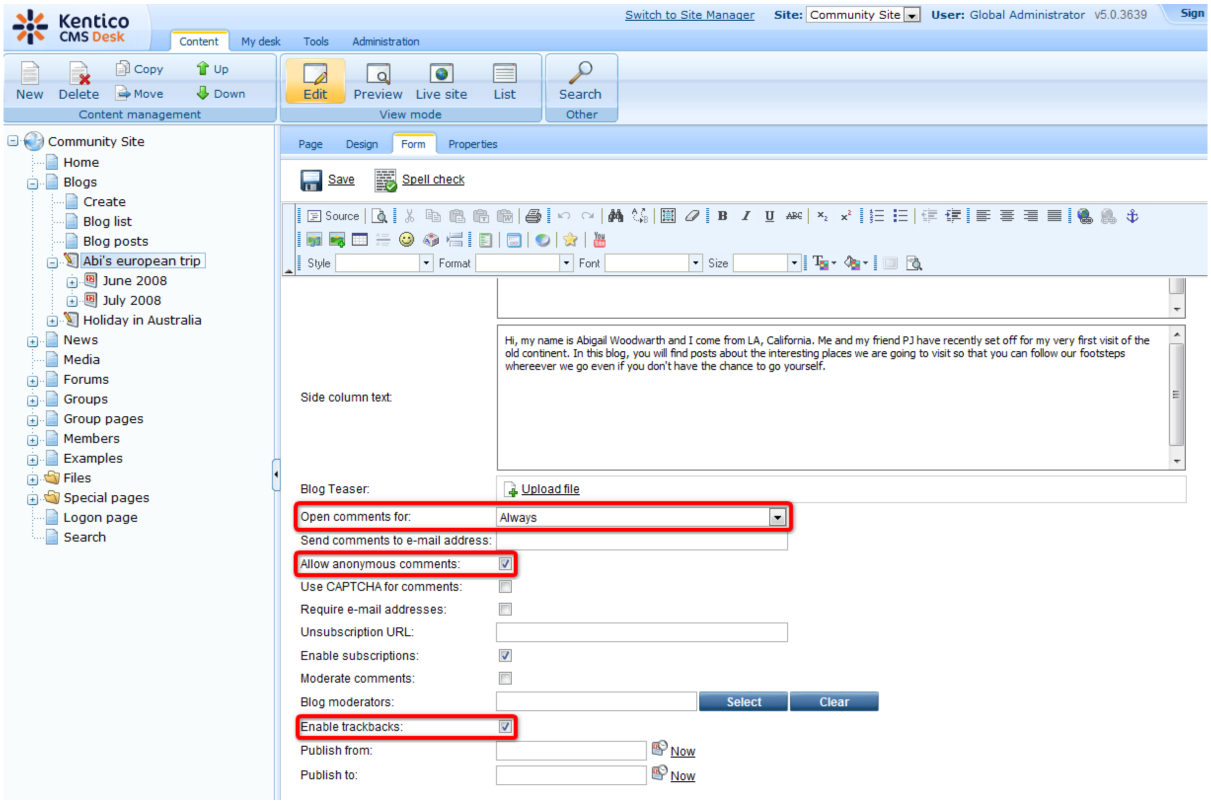
3. When the blog post is submitted, trackback ping is sent to the trackback URLs. If everything is configured correctly at the other blogs, a blog comment is added with the blog post link, title and summary, as you can see in the screenshot below.



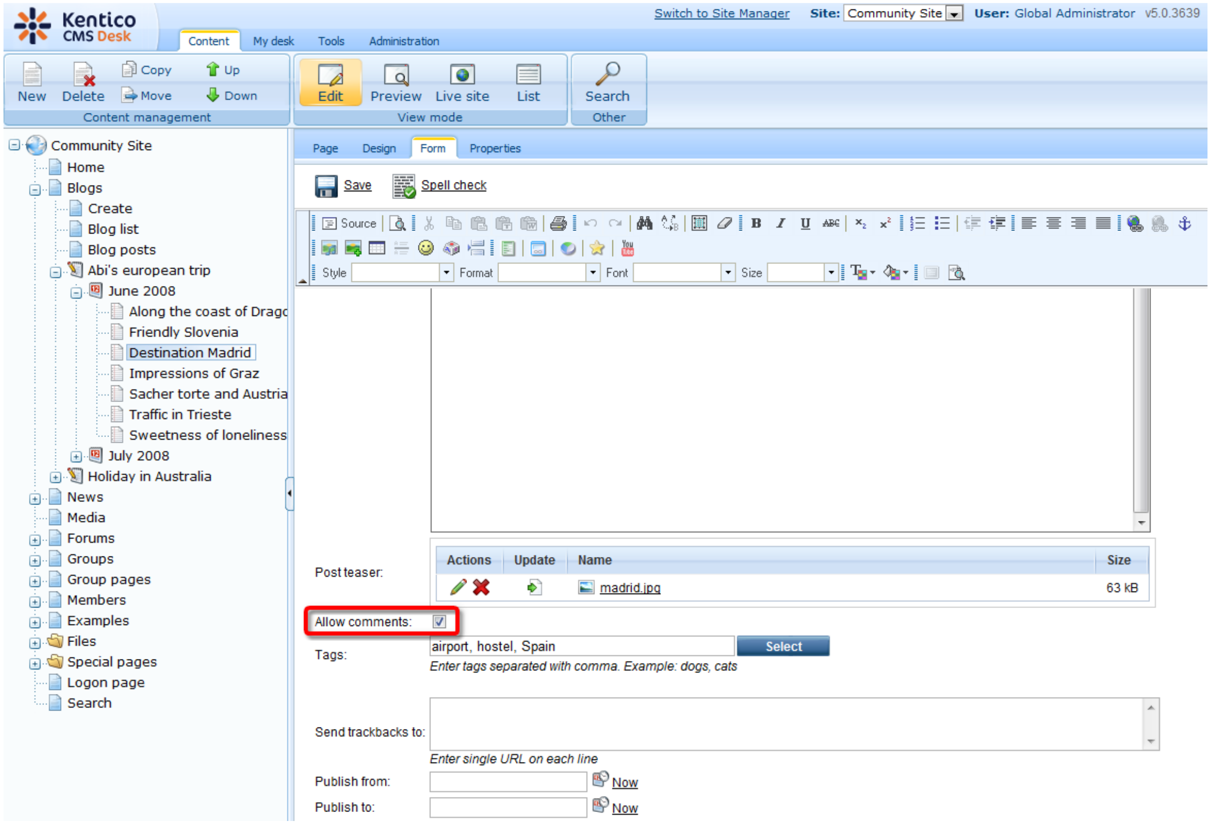
8.9.11.2 Enabling trackbacks

For the trackbacks to be functional in a particular blog, you need to enable the following options on the blog's **Form** tab:

- **Enable trackbacks** - this option enables displaying of the trackback URL with each blog post and inserting of trackbacks as blog comments
- **Allow anonymous comments** - trackback comments are in fact anonymous comments, so this option needs to be enabled for the comments to be inserted
- **Open comments for** - this option defines how long after the insertion of the blog post will inserting of blog comments be possible; trackbacks can also be inserted only within this period



The **Allow comments** option must also be enabled on the pinned blog post's **Form** tab so that the comment can be inserted.



8.9.12 Blog comments notifications

8.9.12.1 Who can be notified

When a new blog comment is added, notification e-mails can be sent to:

- **Blog owners**
- **Blog moderators**
- **Subscribers**

The text below explains to whom notification e-mails are sent under specific conditions:

New comment

1. has been added by the blog owner, blog moderator, user with the **Manage permission** or global administrator

- the comment is marked as APPROVED
- the e-mail is sent to: the blog owner (if their e-mail address is set in blog properties) and the subscribers

2. has been added by anybody else
a) the blog is moderated

- the comment is marked as NOT APPROVED
- the e-mail is sent to: the blog owner (if their e-mail address is set in blog properties) and the

moderators

b) the blog is not moderated

- the comment is marked as APPROVED
- the e-mail is sent to: the blog owner (if their e-mail address is set in blog properties) and the subscribers

Comment editing

1. the comment has been switched from NOT APPROVED to APPROVED

- the e-mail is sent to: the subscribers

2. other comment changes

- no e-mails are sent

8.9.12.2 User subscriptions

Users can subscribe to receiving notifications about new blog comments at some blog post. It can be done in two ways:

1. Users can subscribe when leaving a comment, by checking the **Subscribe me to this blog post** check box. In this case, notifications will be sent to the e-mail address specified in the **E-mail** field above.

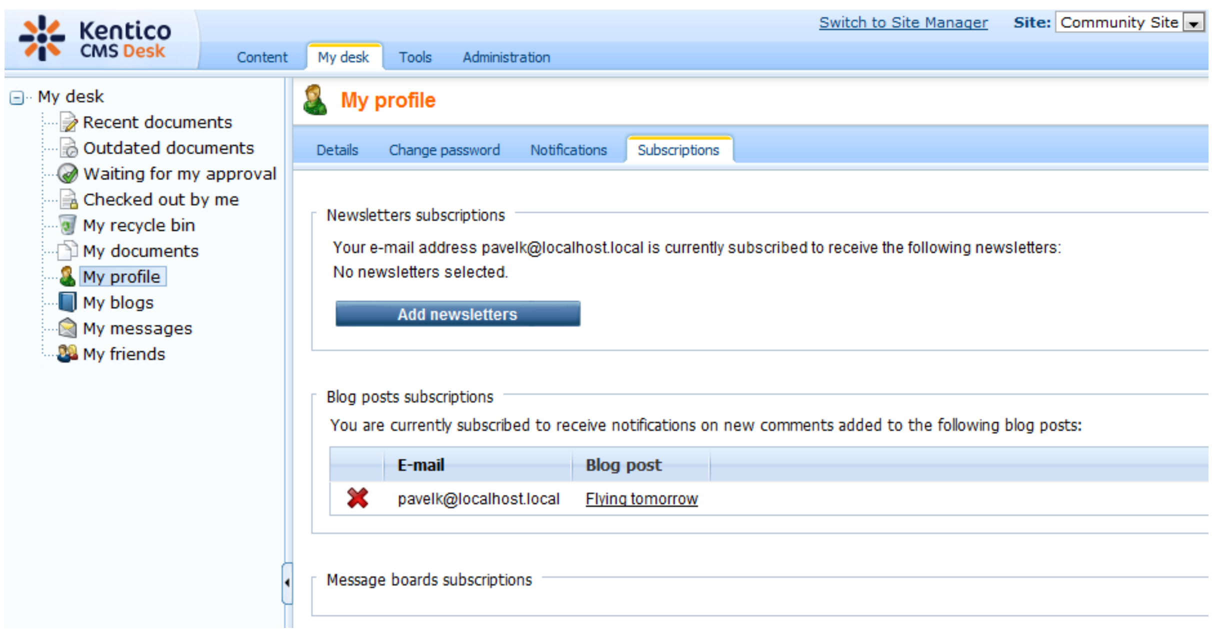
2. Users can also subscribe without leaving a comment, by clicking the **Subscribe** link at the top of the **Leave comment** form.

This displays a subscription form. By entering the e-mail address to the **Your e-mail** field and clicking the **Subscribe** button, users can subscribe too.

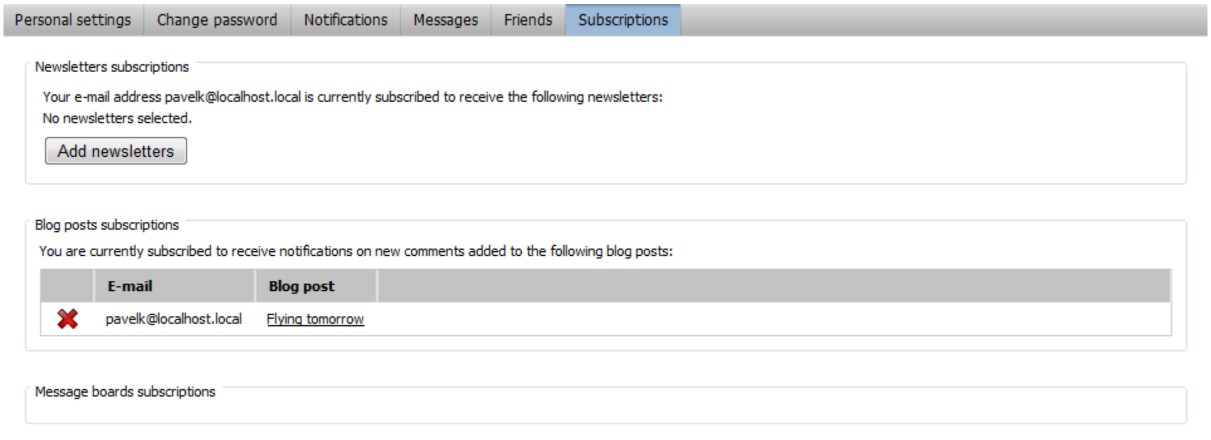
Subscriptions management

Users can view their subscriptions and eventually **unsubscribe** using the **Delete** () icon at the following two places:

1. Users with access to **CMS Desk** can view their subscriptions on the **My Desk -> My profile -> Subscriptions** tab.

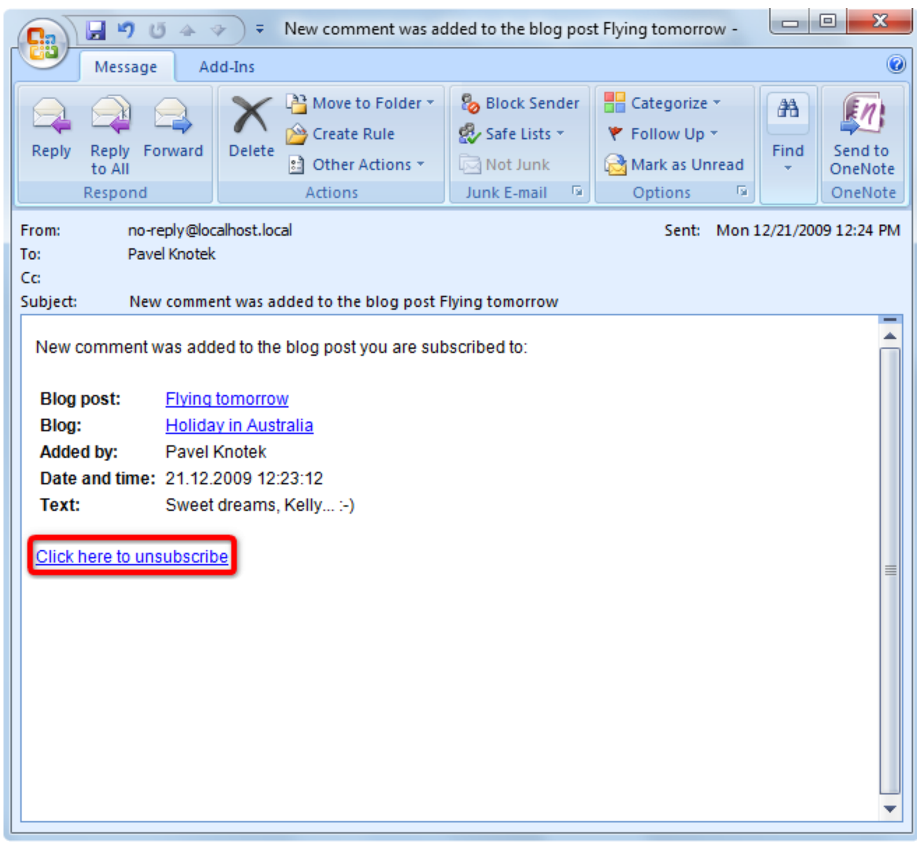


2. On live site, users can view their subscriptions in the **My account** web part. The **Display my subscriptions** property of the web part must be enabled for this to be possible.



Unsubscription possibilities

Subscribers can unsubscribe by clicking the unsubscription link, which is present in each notification e-mail.



For this to work, you have to do the following two things:

1. Place the **Blog post unsubscription** web part to some page. It is recommended to create a special page for this purpose, as you can see at **Community site -> Special-pages -> Blog unsubscribe**. You can set only one specific property of the web part - **Confirmation text -** this is the text that will be

displayed after successful unsubscription.

2. Set the URL of the page created in step 1 as the **Unsubscription URL** property of the blog. This can be done two ways:

- In **Site Manager -> Settings -> Blogs**, by settings the **Blog unsubscription URL** property. This is the default value that will be used by default, if no other URL is set.

- If some different URL is set in the option mentioned above, you can set the value of the **Unsubscription URL** property on the blog's **Form** tab. This value overrides the one set in **Site Manager -> Settings -> Blogs**.

8.9.12.3 E-mail templates

There are three different e-mail templates that can be used when sending notifications about new blog comments, depending on the recipient of the notification:

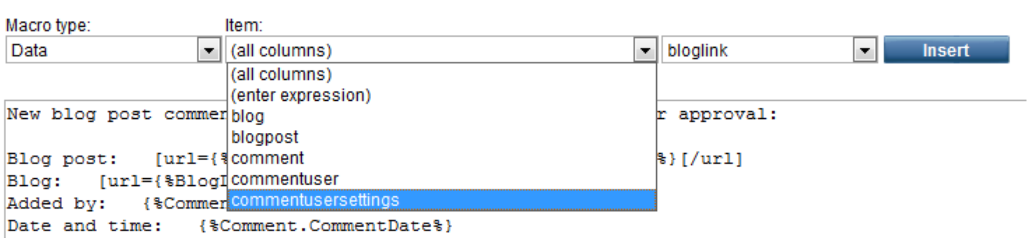
- **Blog owners** - e-mails are based on the **Blogs - Notification to blog owner** e-mail template
- **Blog moderators** - e-mails are based on the **Blogs - Notification to blog moderators** e-mail template
- **Subscribers** - e-mails are based on the **Blogs - Notification to blog post subscribers** e-mail template

The following macros can be used in the e-mail templates:

Data macros

- **Blog.XXX** - where XXX represents a column of the *CONTENT_Blog* table or the *CMS_View_Tree_Joined* view
- **BlogPost.XXX** - where XXX represents a column of the *CONTENT_Blog* table or the *CMS_View_Tree_Joined* view
- **Comment.XXX** - where XXX is a column of the *CONTENT_BlogComment* table
- **CommentUser.XXX** - where XXX is a column of the *CMS_User* table
- **CommentUserSettings.XXX** - where XXX is a column of the *CMS_UserSettings* table

Example: {%CommentUser.Email%}



Source macros

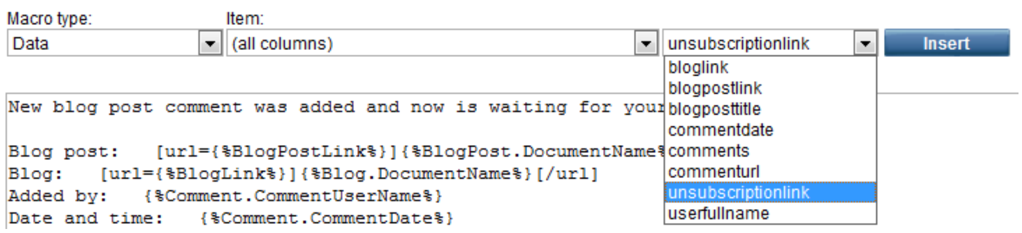
a) Links

- **BlogPostLink** - blog post link
- **BlogLink** - blog link
- **UnsubscriptionLink** - unsubscription link

b) Others - these macros are present due to backward compatibility purposes

- **UserFullName** - the same result as *{%Comment.CommentUserName%}*
- **CommentUrl** - the same result as *{%Comment.CommentUrl%}*
- **Comments** - the same result as *{%Comment.CommentText%}*
- **CommentDate** - the same result as *{%Comment.CommentDate%}*
- **BlogPostTitle** - the same result as *{%BlogPost.BlogPostTitle%}*

Example: *{%BlogPostLink%}*



8.9.13 MetaWeblog API

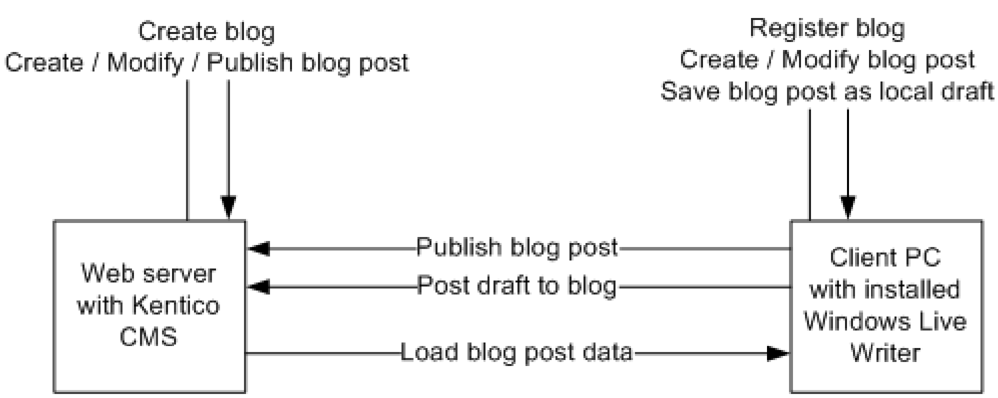
8.9.13.1 Overview

MetaWeblog API is a programming interface that enables external programs to get and set the text and attributes of blog posts. The API uses the XML-RPC protocol for communication between client applications and the blog server.

MetaWeblog API integration enables bloggers to write their blog posts using an application like Windows Live Writer (WLW) or Microsoft Word 2007 (or higher) on their local computer, even when they are offline. Then it enables transfer of these blog posts to the Kentico CMS web application, either by publishing them directly on the live site or just saving them as a draft to be published later.

How it works

The diagram below shows which actions can be performed in both Kentico CMS web application and Windows Live Writer, and the blog post transfer possibilities between the two applications.



The following actions need to be taken in order to use Windows Live Writer to write Kentico CMS blog posts:

1. User creates a blog in Kentico CMS (either via CMS Desk or on the live site).
2. User installs Windows Live Writer and registers their new blog in it.
3. Now the user is ready to create new and modify existing blog posts directly from WLW and transfer the changes into Kentico CMS.

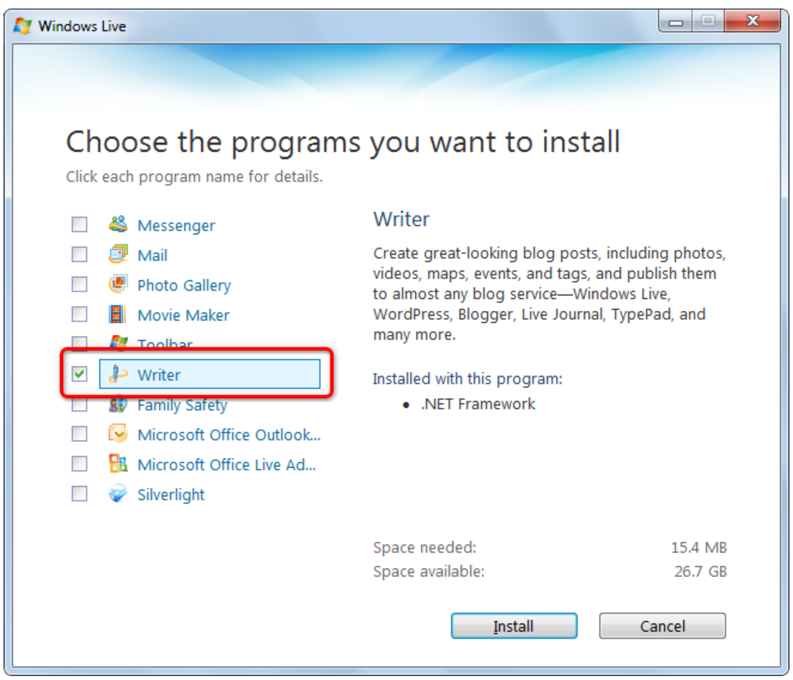
In the **MetaWeblog API** subchapter, you can also learn how to publish a new blog post created with Windows Live Writer to a Kentico CMS Blog, how to manage blog posts and different cultural versions of your blog and finally, how to adjust the settings and security options related to **MetaWeblog API**.

8.9.13.2 Windows Live Writer installation

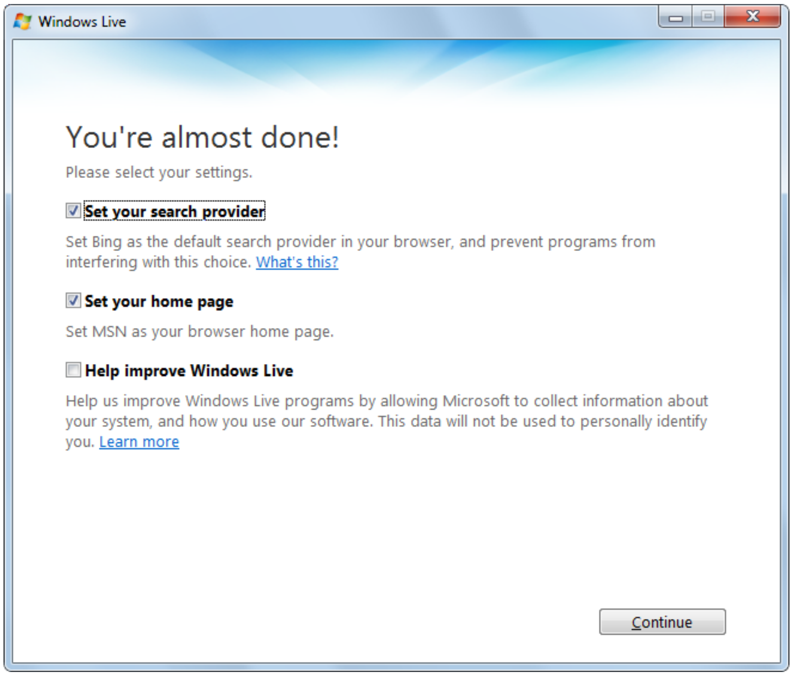
The following steps need to be taken in order to install Windows Live Writer on your machine:

1. Download Windows Live installer from http://download.live.com/writer.

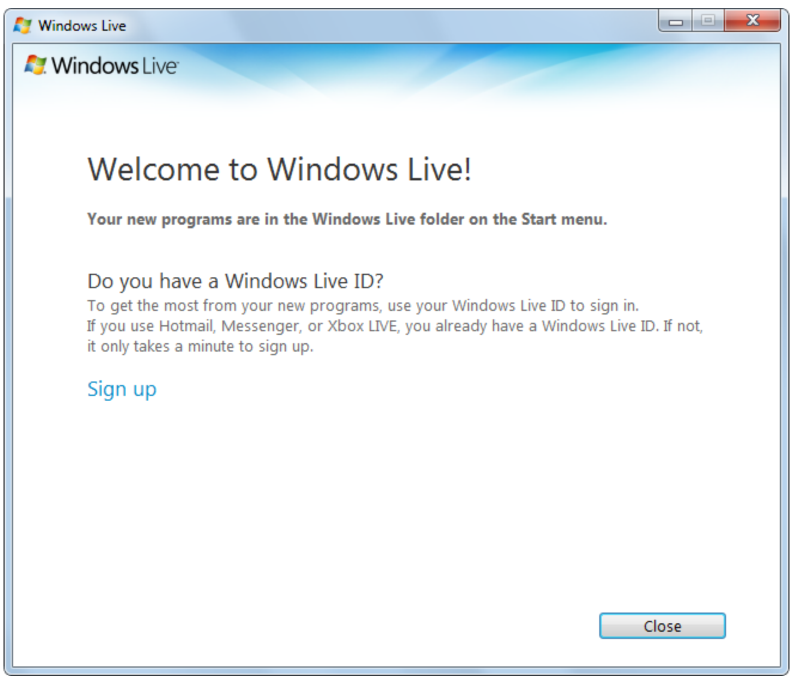
2. Execute the installer. Choose the **Writer** option from the list of available programs. Other options are not directly related to the Windows Live Writer. Choose as you wish and click **Install**.



3. When the installation finishes, it offers you some additional settings, which are not directly related to Windows Live Writer. Choose as you wish and click **Continue**.



4. You will get to the final screen, where you are offered to sign up for a Live ID. This again is not directly related to Windows Live Writer, so do as you wish and finally click **Close** to finish the installer.

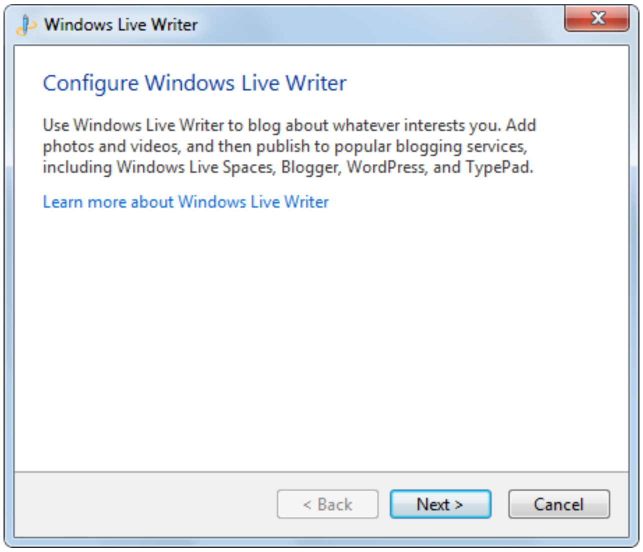


8.9.13.3 Registering blog account

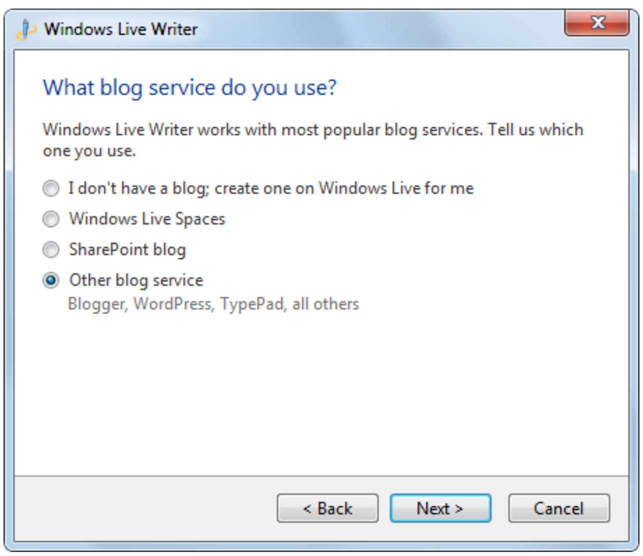
When you run Windows Live Writer for the first time, a wizard will appear, letting you configure connection to your online web service. After the initial setup, the wizard is still accessible under the **Tools -> Accounts -> Add** or **Blogs -> Add blog account** menu options.

The following text describes how to connect Windows Live Writer to a blog in a Kentico CMS instance.

1. The first step of the wizard is just informational. Besides some very basic info, it offers you the *Learn more about Windows Live Writer* link, which redirects you to some in-depth info about the program. Follow it if you wish and click **Next** to proceed further.



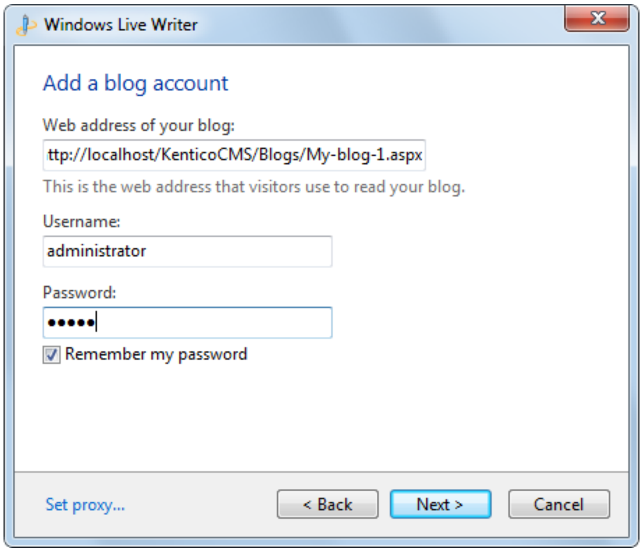
2. This step requires you to select the type of blog service. Select **Other blog service** and click **Next**.



3. Now you need to enter the URL of the blog and the logon credentials of the blog owner. Enter the following details:

- **Web address of your blog**: full Live URL of the blog's title page, e.g. *http://localhost/KenticoCMS_0322/Blogs/My-blog-1.aspx*
- **Username**: user name of the blog owner, e.g. *administrator*
- **Password**: blog owner's password (blank passwords are not supported), e.g. *12345*
- **Remember my password**: enable this option if you don't want to enter the password each time you use the application

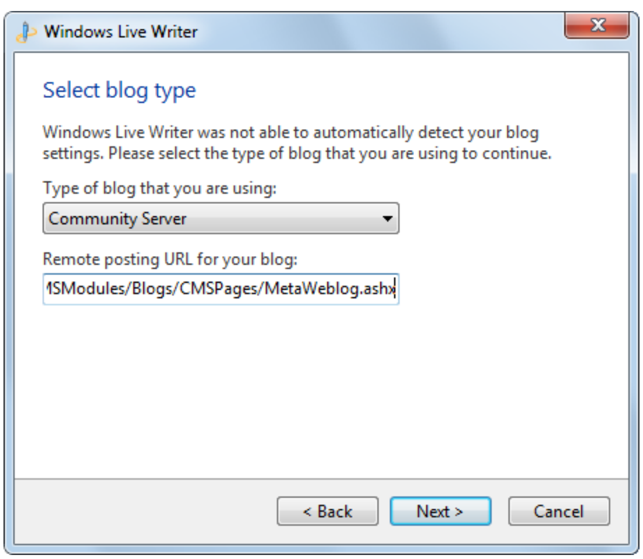
Click **Next** to proceed.



4. In the fourth step, you need to select the type of the blog and the posting URL of the blog. Enter the following details:

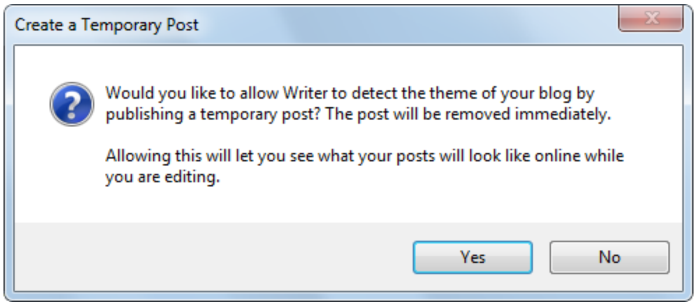
- **Type of blog that you are using**: *Community Server* or *MetaWeblog API* (the first one provides some extra functionality in WLW)
- **Remote posting URL for your blog**: URL of the MetaWeblog API handler, which is always in format *http://<application path>/CMSModules/Blogs/CMSPages/MetaWeblog.ashx* (e.g. *http://localhost/KenticoCMS/CMSModules/Blogs/CMSPages/MetaWeblog.ashx*)

Click **Next**.

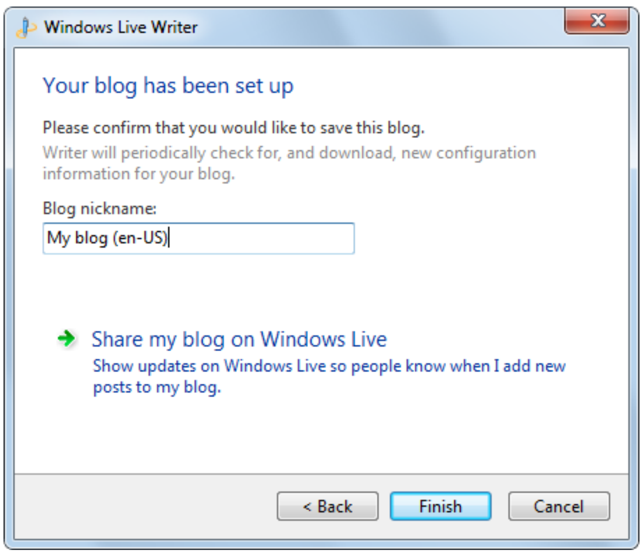


5. Now a dialog pops up, asking you if you want to detect the theme used for the blog so that you can see how the blog post will look like live when you are editing it. Click **Yes** to perform this action. If you click **No**, you can still do it later manually by choosing the **View -> Refresh theme** option from

Windows Live Writer menu.



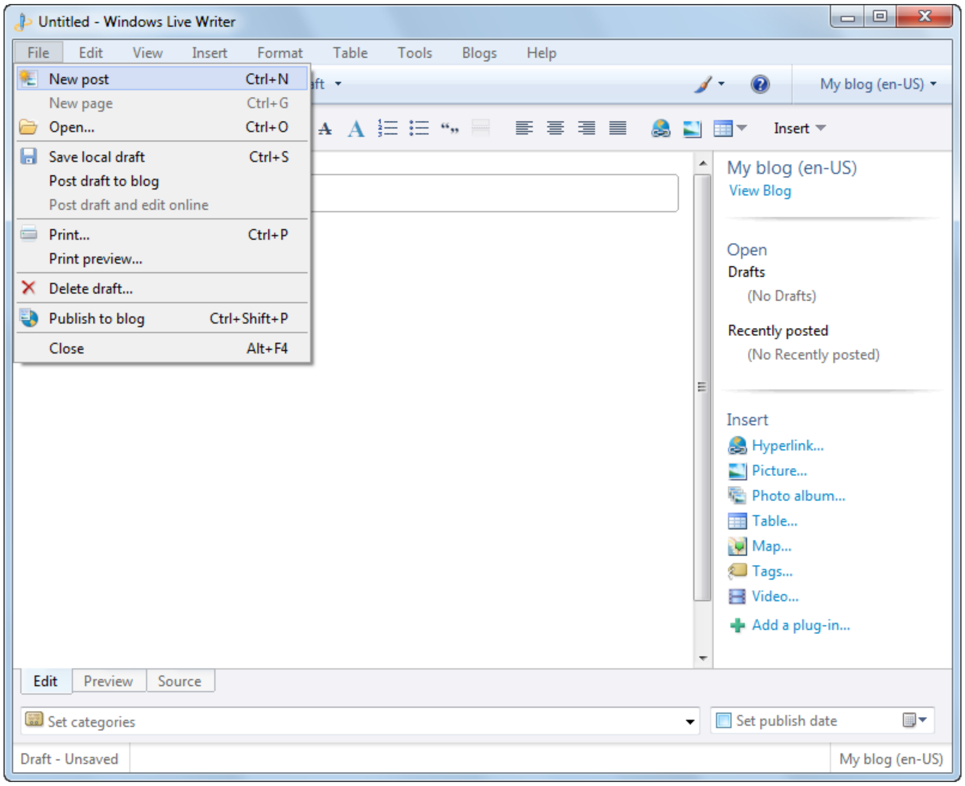
6. In the final step, you just need to enter the **Blog nickname**, which is just a name under which the blog will appear in WLW user interface. Enter whatever you wish and click **Finish**. You blog is now registered and you can start blogging from within Windows Live Writer.



8.9.13.4 Publishing first blog post

In the following example, you will learn how to create a new blog post with Windows Live Writer and publish it to the Kentico CMS blog registered in the previous example.

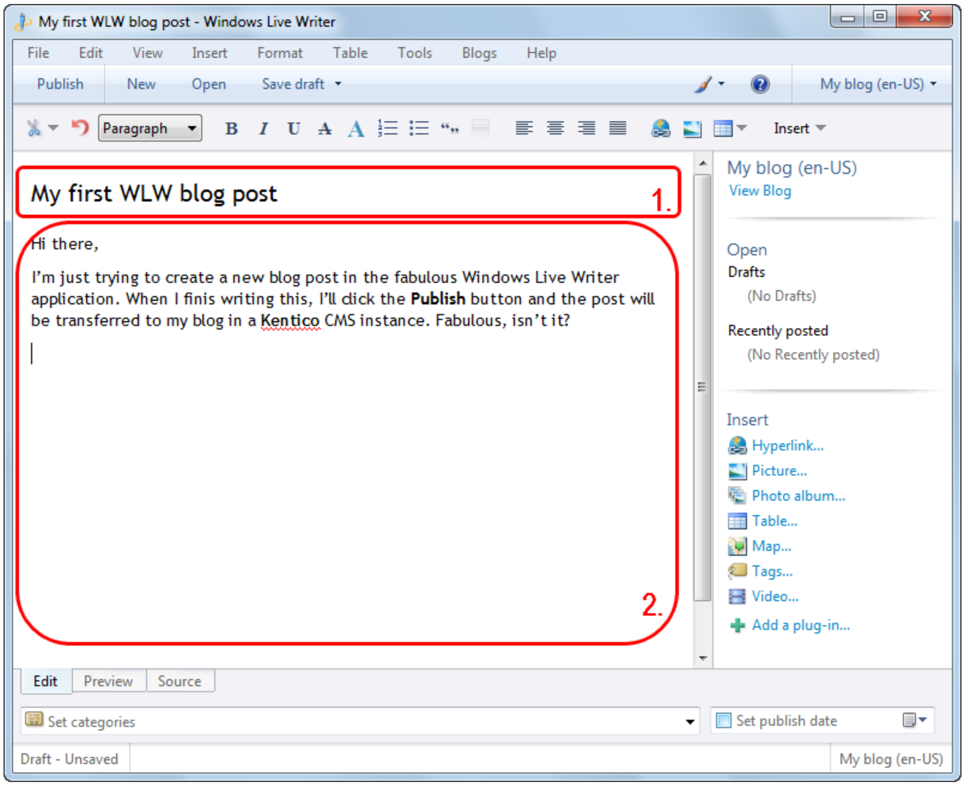
1. Open Windows Live Writer and select **New post** from the **File** menu (or click **Ctrl+N**).



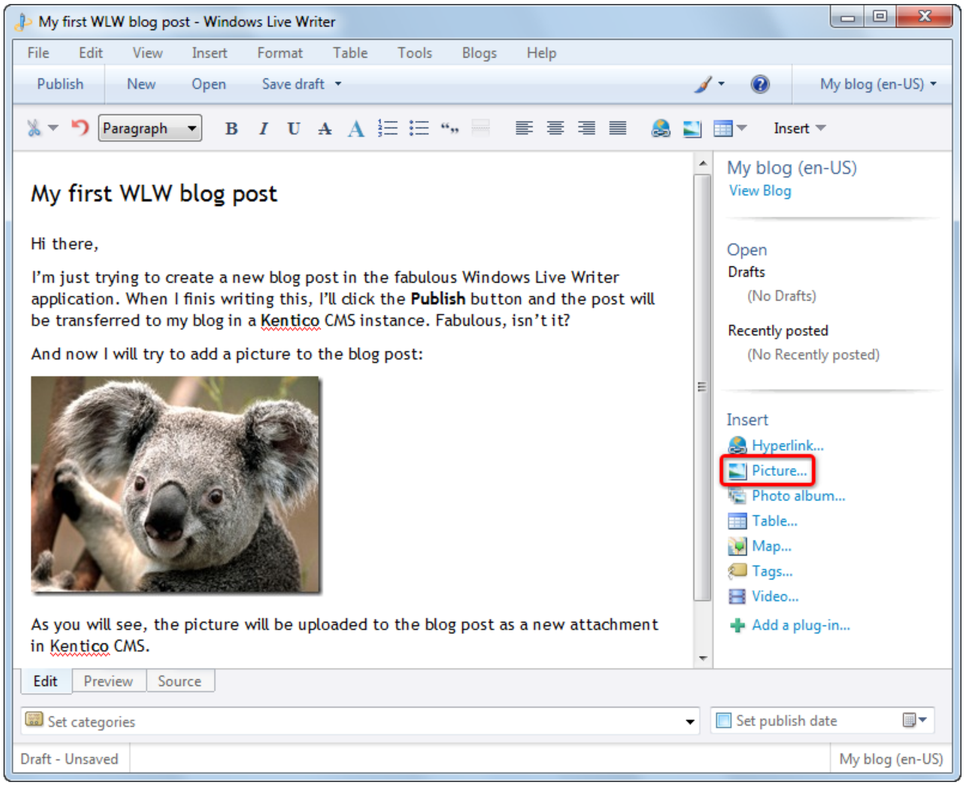
2. The gray rectangle at the top (1.) is where you enter the title of the blog post. What you enter here will be used as a value of the **Post title** field in your Kentico CMS blog post.

The text area below (2.) is where you enter the text of the blog post itself. This will be used for the **Post text** field, and in case you enabled the automatic summary (learn here how to do it), its beginning will be used for the **Post summary** field too.

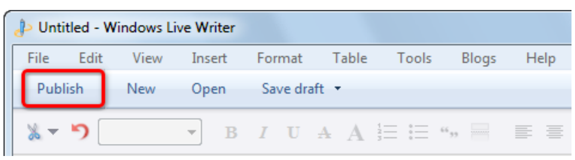
Enter some title and blog post text.



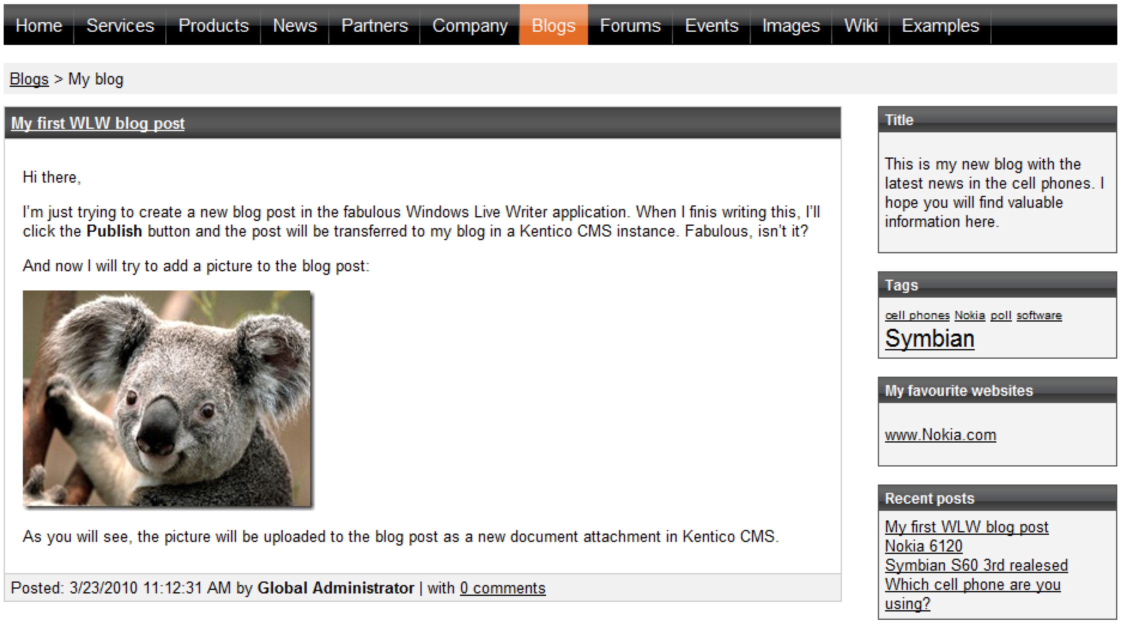
3. Now let's try adding an image. It is quite easy with Windows Live Writer, just click the **Picture...** option in the **Insert** menu on the right and select some image from your local disk. You should achieve that your blog post looks somewhat similar to the screenshot below.



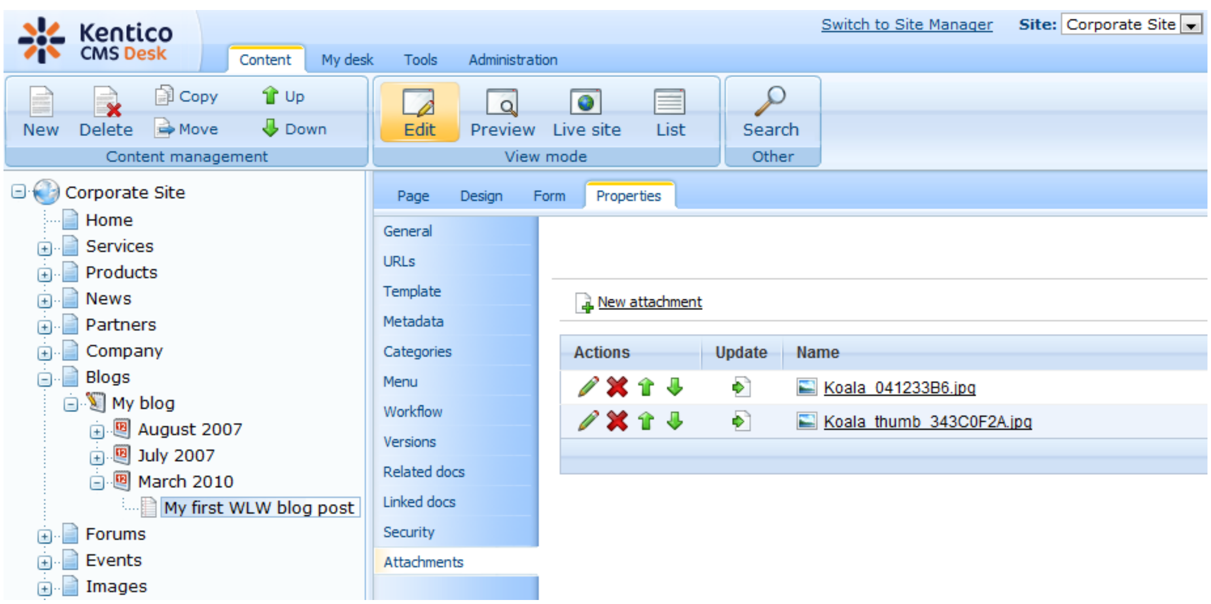
4. Now that your blog post is ready, we can publish it to the live site. Click the **Publish** button in the top toolbar.



5. When the publishing finishes, your blog will be displayed in your browser and you will see the new post published.



6. If you view the blog post in CMS Desk and switch to its **Properties -> Attachments** tab, you will see that the image has been uploaded as an attachment of the post. There are two versions of the image - the first one is the original full-size image, while the second one is its thumbnail used in the post text (e. g. in the screenshot above).

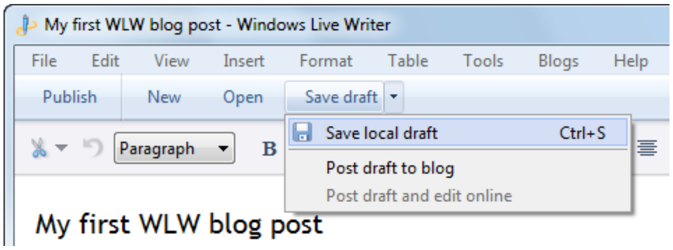


8.9.13.5 Managing blog posts

This chapter describes some additional options that you have when creating or editing a blog post.

Save local draft

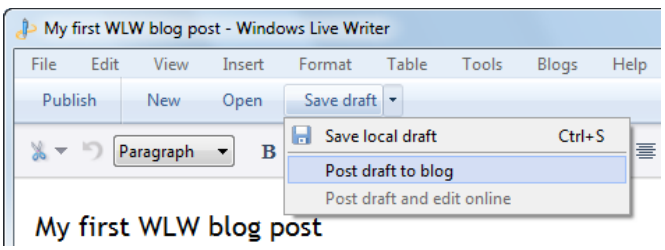
Current local version of the blog post is saved as blog post local draft; each blog post can have only one local draft. This action doesn’t propagate any changes into the CMS.



Post draft to blog

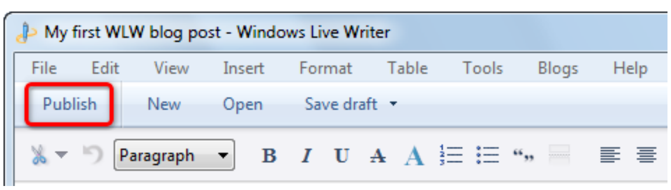
Current local version of the blog post is posted to blog, but its local draft is not removed. Depending on workflow settings, the following two situations can occur in the CMS:

a) **Blog post document is under workflow -** new minor version of the blog post is created (posted version is not published)
b) **Blog post document is not under workflow -** blog post is published immediately; this means that this action is identical to the *Publish* action, i.e. the *Post draft to blog* action makes sense only in case that workflow is enabled for blog post documents



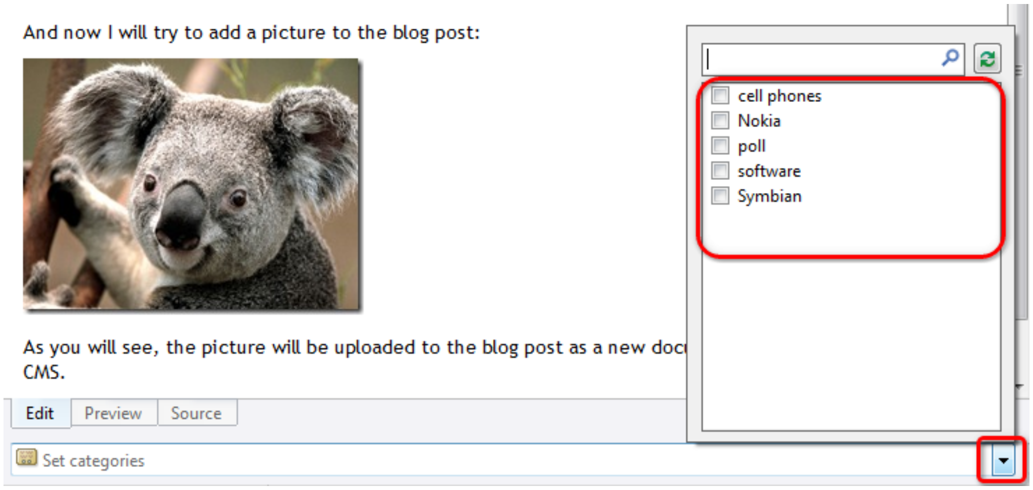
Publish

Current local version of the blog post is posted to blog and published immediately. Local draft of the published blog post is automatically removed.



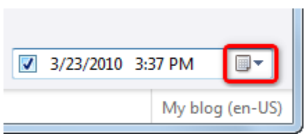
Tagging

The tag selector in WLW enables you to tag the blog post with tags from the parent blog's Kentico CMS tag group. The tag group of the blog post itself is not reflected here at all.



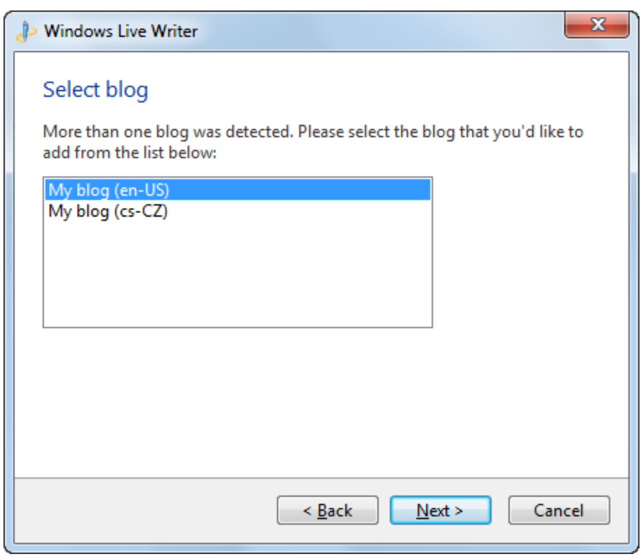
Publish date

You can specify a date here, which will be propagated into the **Publish from** field on the **Form** tab in CMS Desk. The value determines the date and time when the blog post gets published on the live site. It will be converted to the user's time zone before it is set as the **Publish from** property of the blog post in Kentico CMS. If no time is set here, the blog post is published immediately.



8.9.13.6 Multilingual support

In case that your blog exists in more than one cultural version, you will get the following dialog displayed after blog type selection (step 4) when registering your blog account.

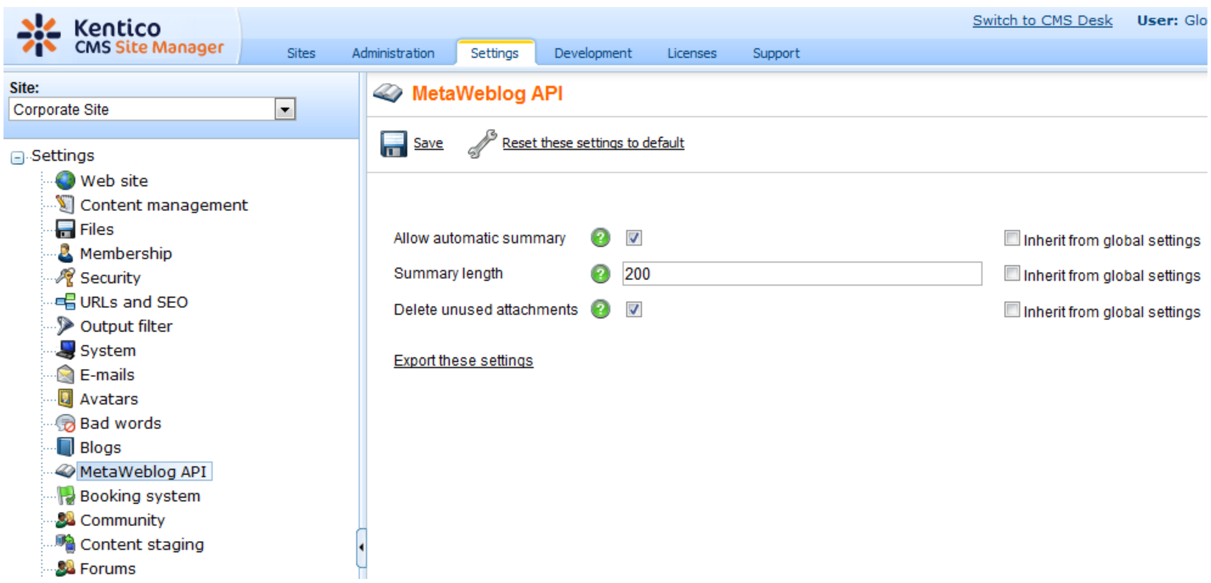


In the dialog, you need to choose which cultural version will the account be registered for. This means that you need to register **one dedicated account for each cultural version** of your blog.

8.9.13.7 Settings

Settings related to MetaWeblog API can be performed in **Site Manager -> Settings -> MetaWeblog API**. The following settings can be configured:

- **Allow automatic summary**: if enabled, blog post summary will be created automatically from the beginning of the blog post text, and will be as long as the value of the Summary length property; this is applied only when a post is created - when editing an existing post, the summary is not updated
- **Summary length**: length of automatic blog post summary in characters
- **Delete unused attachments**: if enabled, all blog post attachments which are not used in post text or have not been uploaded via WLW (i.e. have been uploaded via Kentico CMS user interface) will be deleted when an existing blog post is modified and then published from WLW



8.9.13.8 Security

Managing blog posts

The following text explains what conditions need to be met in order to perform the listed actions in Windows Live Writer. The conditions need to be valid for the user account used to access the blog (as entered in step 3 here).

To **view a blog post**, the user needs to:

- be the owner of the blog post
- or have the the **Read** document permission for the blog post granted (as explained here)
- or be a global administrator

To **modify a blog post without workflow**, or to **modify a blog post under workflow without custom workflow steps**, the user needs to:

- be the owner of the blog post

- or have the **Modify** document permission for the blog post granted (as explained here)
- or be a global administrator

To **modify a blog post under workflow with custom workflow steps**, the user needs to be:

- member of the role which can approve the document in the custom workflow step (see Defining a workflow for more details)
- or a global administrator

To **delete a blog post**, the user needs to:

- be the owner of the blog post
- or have the **Delete** document permission for the blog post granted (as explained here)
- or be a global administrator

Banned IPs

If the user's IP is on the Banned IPs list, the user will not be able to transfer any data between Kentico CMS and the client application (WLW,Microsoft Word 2007+, ...).

8.10 Booking system

8.10.1 Overview

The Booking system module allows you to manage events and their attendees. You can use it for both on-line and off-line meetings.

You can find the Booking system module in **CMS Desk -> Tools** section.

8.10.2 Publishing the events

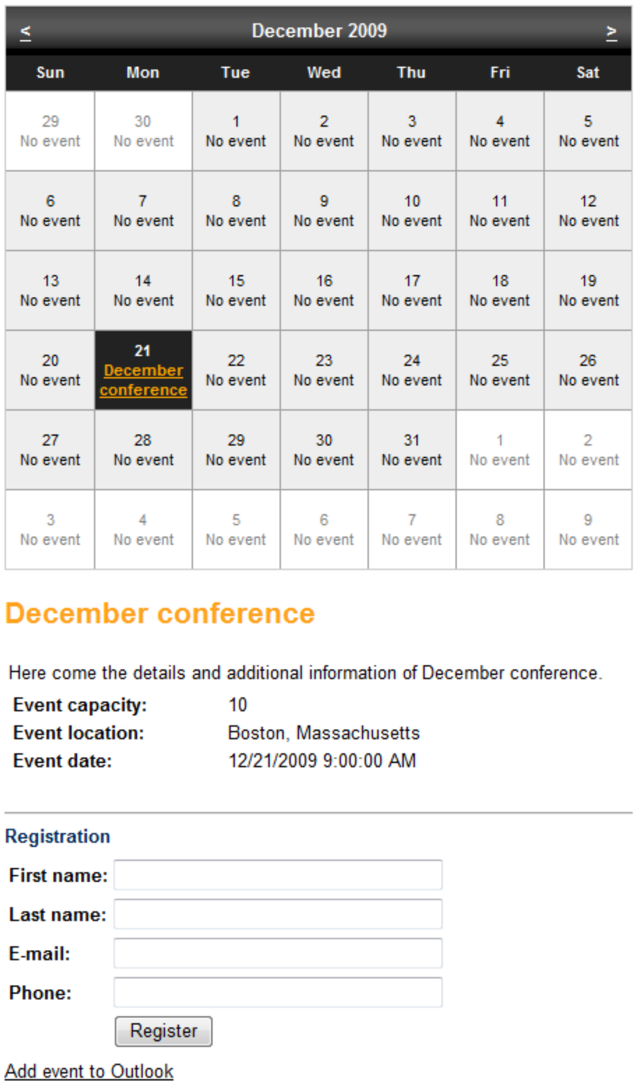
The events managed by the Booking system are actually documents of type **Event (booking system)**. The Event document uses the following fields:

Event name	Name
Event summary	Summary
Event details	Details
Event location	Location
Event date and time	When the event starts
Capacity	Maximum number of attendees
Allow registration over capacity	Indicates if the number of registered attendees can be higher than the capacity.
Open from	Date and time when the registration starts
Open to	Date and time when the registration ends

If you wish to publish events, you can use the page template **Events/Event calendar with event**

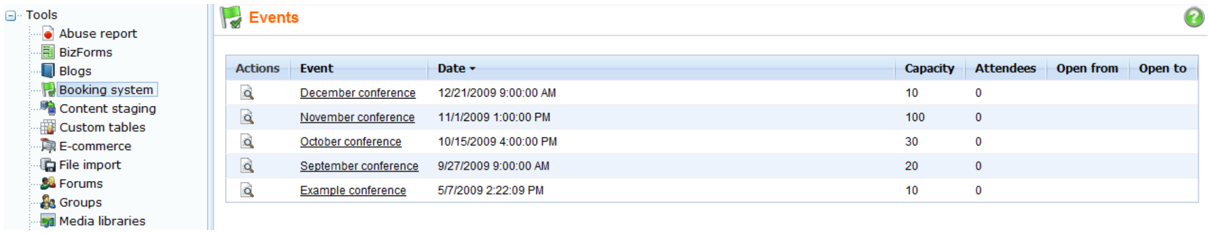
details to display the event calendar and allow users to register or you can build your own page template.

The event calendar is displayed using the **Events -> Event calendar** web part. The registration is ensured by the **Booking system/Event registration** web part. It is displayed when the event details are selected and allows visitors to sign up for the event.



8.10.3 Managing attendees

The attendees who register for the event can be managed in the **CMS Desk -> Tools -> Booking system** section. Here you can see the list of all events, their capacity and number of attendees.



When you view some event, you can see the registered attendees and edit their details or remove them. You can also resend the invitation e-mail.

Sending e-mail to all attendees

On the **Send e-mail** tab, you can send an e-mail to all attendees and inform them about event changes, etc.

8.10.4 Security

The security is split into two sections:

Management of events

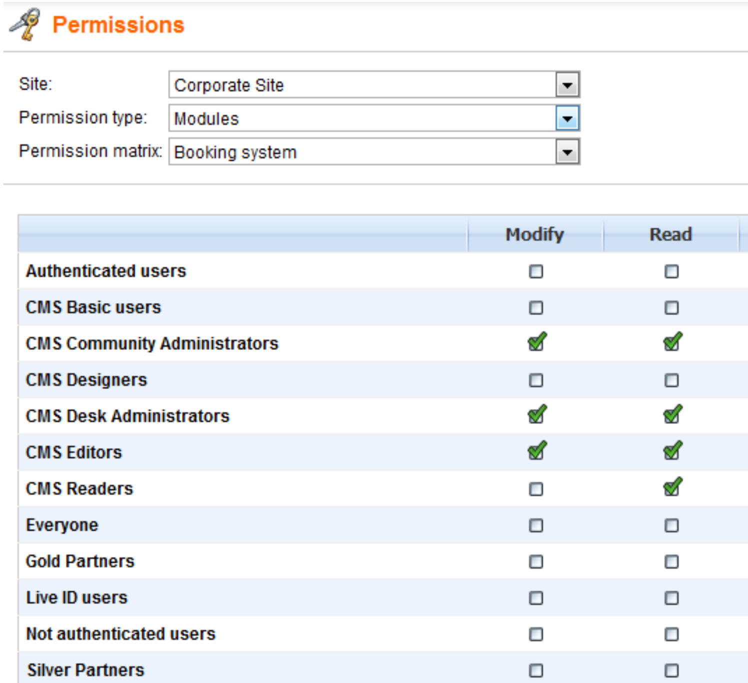
Since the events are standard documents, the users who manage them need to have appropriate document permissions as described in Permissions for modules and documents.

Management of attendees

The attendees can be managed in the **Tools -> Booking system** module. The permissions for this module in the **Administration -> Permissions** section need to be granted to the users who want to perform the management.

You can assign the following permissions to the roles:

- **Modify -** allows users to modify (add, update, delete) the list of attendees and their details
- **Read -** allows users to read the list of attendees, resend messages and send mass e-mail



8.10.5 Booking system internals and API

8.10.5.1 Database tables and API classes

The Booking system module uses the following database table:

- **CONTENT_BookingEvent** - the CMS.BookingEvent is a standard Kentico CMS document type, therefore, it has this associated database table where information about particular booking events is stored
- **Events_Attendee** - this table is used to store information about booking event attendees

The Booking system API is provided by the following **CMS.EventManager** namespace classes:

- **EventAttendeeInfo**, **EventAttendeeInfoProvider -** these classes provide functionality for managing of event attendees

The following topics show how methods from these classes can be used to manage event attendees:

- Adding an attendee
- Editing an attendee
- Removing an attendee

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.10.5.2 Adding an attendee

The following code example demonstrates how an attendee can be added to a booking system event.

[C#]

```
using CMS.EventManager;

int eventNodeId = 10;

// Create new attendee object
EventAttendeeInfo eai = new EventAttendeeInfo();

// Set its properties
eai.AttendeeEventNodeID = eventNodeId;
eai.AttendeeFirstName = "John";
eai.AttendeeLastName = "Smith";
eai.AttendeeEmail = "john.smith@example.com";

// Save the object
EventAttendeeInfoProvider.SetEventAttendeeInfo(eai);
```

8.10.5.3 Editing an attendee

This code example explains how attendee details can be modified using the API.

[C#]

```
using CMS.EventManager;

int eventNodeId = 10;
string email = "john.smith@example.com";

// Get attendee info object
EventAttendeeInfo eai = EventAttendeeInfoProvider.GetEventAttendeeInfo
(eventNodeId, email);

// Update its properties
eai.AttendeeFirstName += "_Updated";

// Save the object
EventAttendeeInfoProvider.SetEventAttendeeInfo(eai);
```

8.10.5.4 Removing an attendee

This code example shows how an attendee can be removed from a booking system event.

[C#]

```
using CMS.EventManager;

int eventNodeId = 10;
string email = "john.smith@example.com";

// Get attendee info object
EventAttendeeInfo eai = EventAttendeeInfoProvider.GetEventAttendeeInfo
(eventNodeId, email);

// Delete the attendee object
EventAttendeeInfoProvider.DeleteEventAttendeeInfo(eai);
```

8.11 Categories

8.11.1 Overview

The Categories module enables users to categorize their documents in a similar way like the Tags module does.

There are two types of categories:

- Global categories - categories created by the global administrator; such categories can be used by all users.
- Custom categories - categories created by a user; only the user who created such categories can use them.

Please note that there are differences between the Categories module and the Tags module. As mentioned above, user defined categories are available only to the users who created them. However, user defined tags are accessible globally.

Any document in the content tree can be assigned to an unlimited number of categories. The Categories module contains the **Category list** web part. This web part can display a list of categories in the form of clickable links. By clicking on a particular category link, a list of all documents in the selected category is displayed.

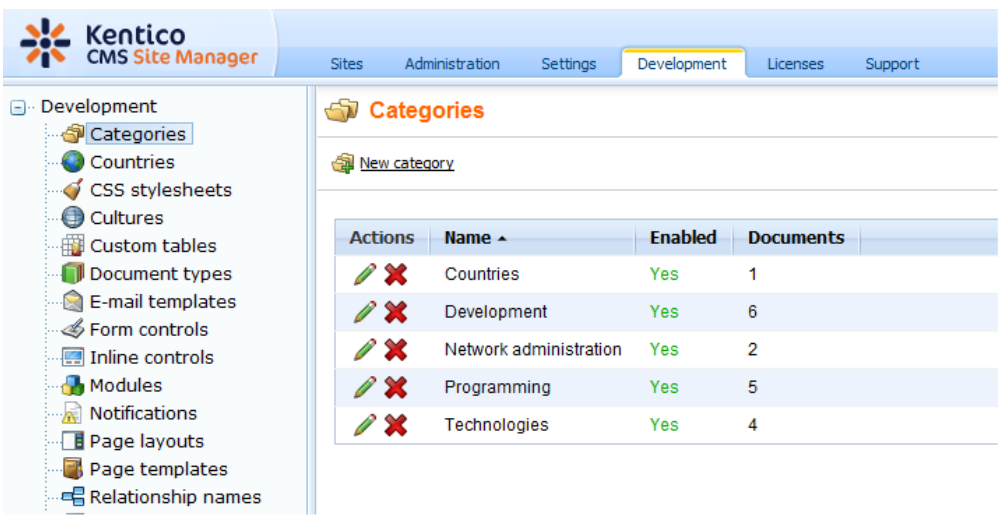
Please note that to be able to create a new category, you must be signed in either as a system administrator or any user granted access to the **Properties** tab in **CMS Desk -> Content -> Edit** section with the **Categories** frame enabled. If you are signed in as user whose UI profile lacks the above mentioned access rights, it is necessary to adjust the respective UI profile. You may need to contact the global administrator to perform this task for you.

- If you would like to add a document to either a global or user defined category, please refer to the Adding a document to a category topic.
- If you would like to learn how to display a list of enabled global and custom categories, please refer to the Using the Category list web part topic.

8.11.2 Global categories

Global categories are created by system administrators and all users can add documents to these categories. Administrators can manage global categories in **Site Manager -> Development -> Categories**. On this page, only the **global categories** can be created and managed.

> **Please note**
>
> Users without administration privileges can also edit global categories created by system administrators (**CMS Desk -> Properties -> Categories**); however, with certain limitations compared to editing here.



To create a new global category, just follow these three easy steps:

1. Click the **New category** link.

2. Enter the required details:

Display name	Display name of the category.
Code name	Code name of the category.
Description	Description text of the category.
Enabled	Indicates if the category is enabled in the system. Documents cannot be assigned to categories that are not enabled.

3. Click **OK** to save the changes. The category has just been created and should be visible in the categories list.

8.11.3 Custom categories

There are two ways to create custom categories. However, only users with administration privileges for the particular site can create custom categories both the ways.

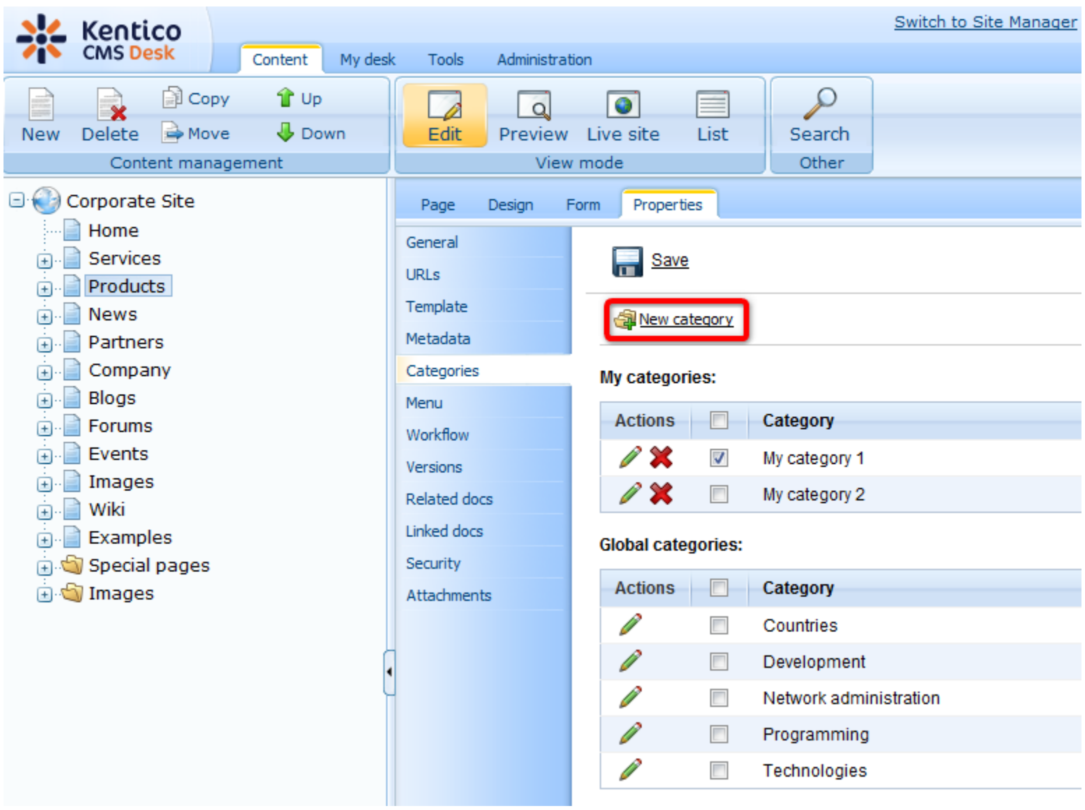
Creating new custom categories

Each user can create their own categories and assign documents to them. This can be done in **CMS Desk**, after selecting a document from the content tree and switching to its **Properties -> Categories** tab.

The current user's **custom categories** are listed in the **My categories** section. These categories can be **Edited** ( ) or **Deleted** ( ) in this section. Global categories are listed below and can also be edited here with certain limitations compared to editing in **Site Manager**.

To create a new custom category, just follow these three easy steps:

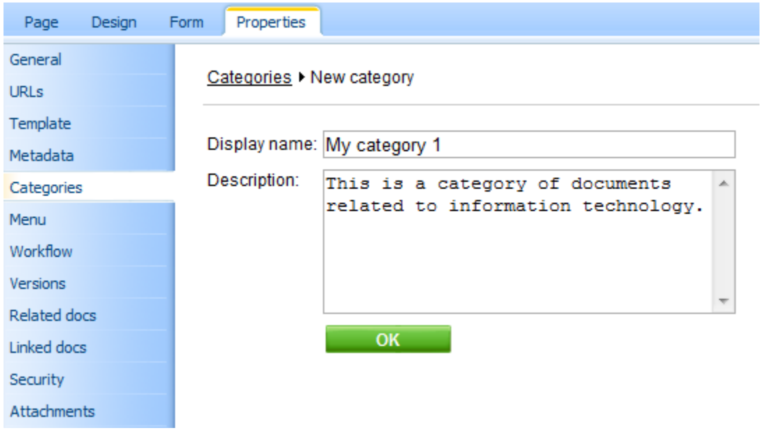
1. Click the **New category** link.



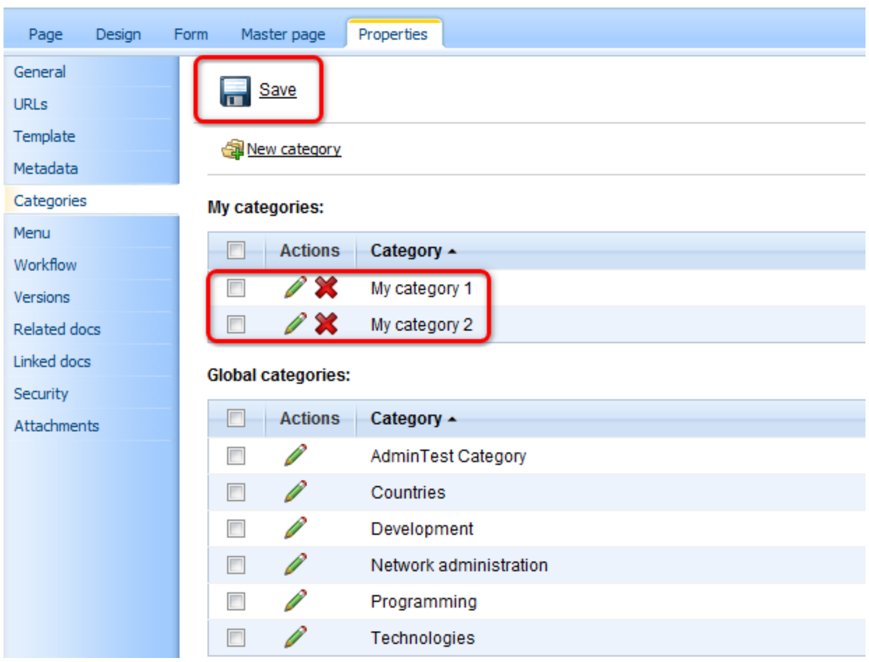
2. Enter the required details:

Display name	Display name of the category.
Description	Text describing of the category.

And click **OK**.

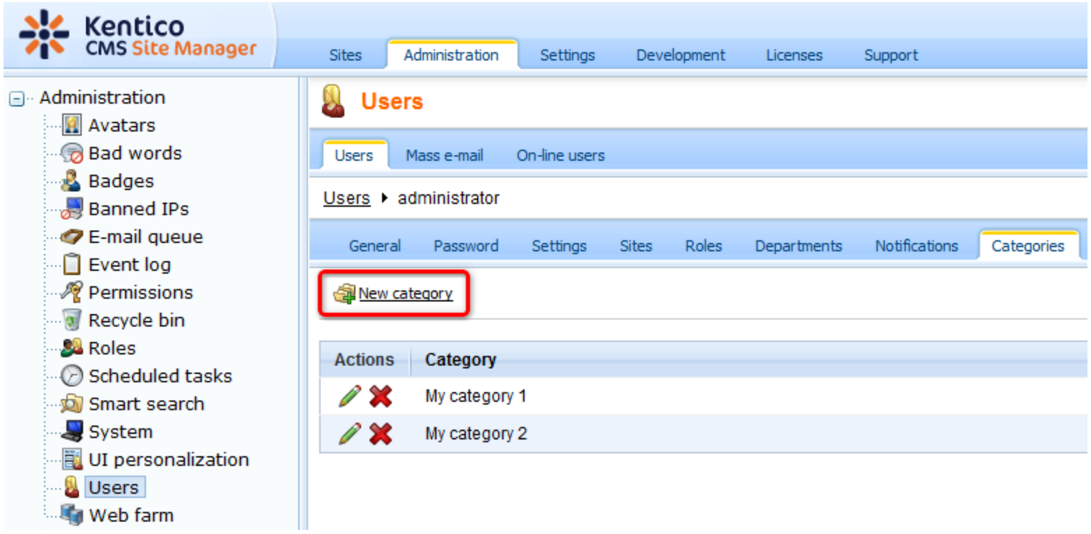


3. You should be redirected back (**CMS Desk -> Properties -> Categories**) to the list where your new custom category should be visible. Click **Save** to confirm the changes you have made.



Administrating users' custom categories

Administrators can manage custom categories of other users in **Site Manager -> Administration -> Users**. Choose to **Edit** ( ) some of the users and switch to their **Categories** tab. You can edit and eventually create the selected user's **custom categories** here. Categories can be created in a similar way as mentioned in the previous examples, by clicking the **New category** link and entering the required details.

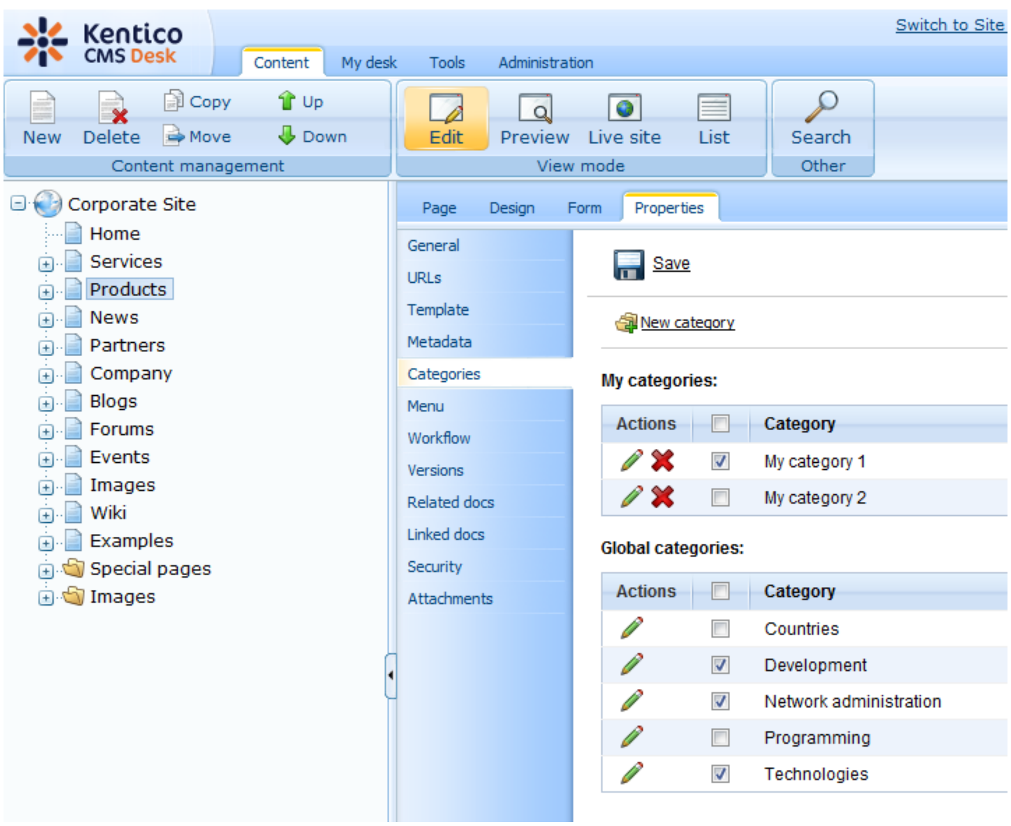


8.11.4 Adding a document to a category

Adding documents to categories is very simple.

1. Log in to **CMS Desk** and in **Edit** mode, select the document from the content tree.

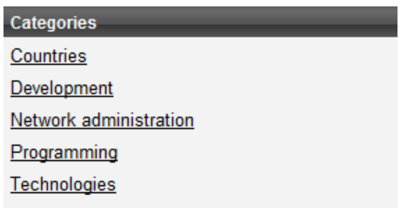
2. Switch to its **Properties -> Categories** tab.



3. In the categories lists, check the appropriate categories' **Select** check-boxes and click **Save**.

8.11.5 Using the Category list web part

The category list web part is used to display a list of enabled global and custom categories. For each of the categories, there is a number of documents in the category displayed in brackets after the category's name. If the site visitor clicks any of the categories, they will be redirected to a page containing a list of links leading to documents in the category.



The Category list always works 'in pair' with some repeater web part. When a category is clicked in the category list, a list of articles in the selected category is displayed in the repeater. The repeater can be located either on the same page like the category list, or on some other page to which the site visitor will be redirected. You can find an example of this behavior on the Corporate Site sample website, in section **Examples -> Web parts -> Tagging & Categories -> Category list.**

For this to work, you have to do the following property settings to the web parts:

Category list:

- The placement of the repeater is defined by the Category list's **Category list page** property. In case the repeater is placed on the same page as the category list, the value should be left blank. In case it is placed on some other page, you should enter the **alias path** of that page.

- ID of the clicked tag is transferred to the repeater in form of a query string parameter. The name of the parameter can be set using the **Querystring parameter name** property. The repeater displays the appropriate list of documents based on the value that it gets via this parameter.

Repeater:

- Set the value of the **Path** parameter to the location in the content tree where the documents are stored.

- Set the value of the **Document types** parameter to the document type(s) that is (are) to be displayed.

- Select the transformations that you want to use for the **Transformation** and **Selected items transformation**.

- Finally, use the following code as a value for the repeater's **WHERE condition** parameter. The **categoryid** value should be replaced by the name set in the Category list's **Querystring parameter name** property.

```
'{?categoryid?}' = '' OR (DocumentID IN ( SELECT DocumentID FROM
CMS_DocumentCategory WHERE CategoryID = '{?categoryid?}' ))
```

8.12 Content rating

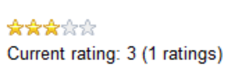
8.12.1 Overview

The content rating module can be used to give users the possibility of rating any document on your website. The module's main part is the Content rating web part. You can place this web part on any page of your website and let site visitors rate the content of the particular page. You can then view the overall rating on the document's Properties tab in CMS Desk.

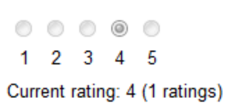
8.12.2 How to enable content rating on the web site

The content rating web part gives users the possibility of rating a document. It has three default appearance modes:

Stars



Radio-buttons



Drop-down

It is also possible to create other appearance modes by creating your **custom controls**. These must be placed in **~/CMSAdminControls/ContentRating/Controls/** and inherit from **ExtendedControls.AbstractRatingControl.cs**, just like the default ones.

You can place the web part on any page of your website to enable users to rate the content of this particular page. In the following example, we will add the web part to a news item on the sample Corporate site.

1. Sign in to **CMS Desk** as *administrator* (blank password by default). Switch to the **Design** mode and select **News -> Your first news** from the content tree.

2. Click the **Add web part** () icon at the top right corner of the **zoneLeft** web part zone. Select the **Content rating -> Content rating** web part and click **OK**.

3. In the web part properties window, you can set the following specific properties:

- **Rating value -** this property can be used to set the displayed rating value explicitly; a value from the <0,1> interval can be used; if blank, the real rating calculated as an average value of particular ratings is displayed
- **Rating type -** appearance of the web part; Stars, Radio-buttons or Drop-down list as depicted above can be chosen
- **Max rating value -** size of the rating scale; e.g. if 7 is entered, rating will be possible on a scale from 1 to 7

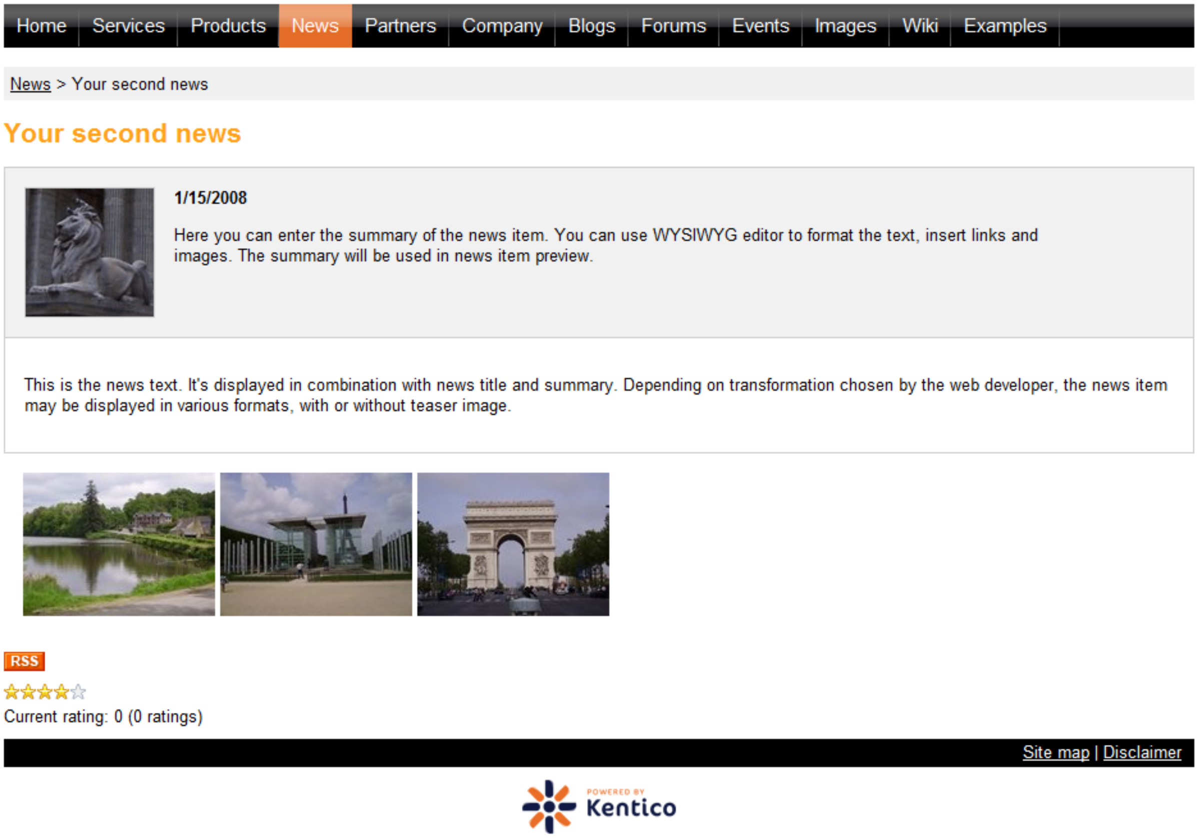
- **Show results -** if checked, overall rating results will be displayed; if unchecked, users can rate, but don't see the results
- **Result message -** message showing overall rating results; the {0} macro shows overall rating (for one decimal rounding, you can use {0:0.#}); {1} displays the total number of votes
- **Message after rating -** text message displayed a user submits their rating; macros that can be used: {0} your rating, {1} overall rating, {2} overall number of votes

- **Check permissions -** if checked, permissions set by the Allow for public and Hide to unauthorized roles properties will be checked
- **Allow for public -** if checked, rating will be allowed to public unauthorized users; if unchecked, all users will be allowed to rate
- **Check if user rated -** if checked, users will be allowed to vote only once; to indicate that the user already voted, Kentico CMS stores a **DocRated** cookie in your web browser, the cookie contains **NodeID**s of the rated documents divided by vertical bars (e.g. |*4*|*61*|*229*|*230*|*228*|*369*|)
- **Hide to unauthorized users -** if checked, the web part will be hidden to unauthorized users

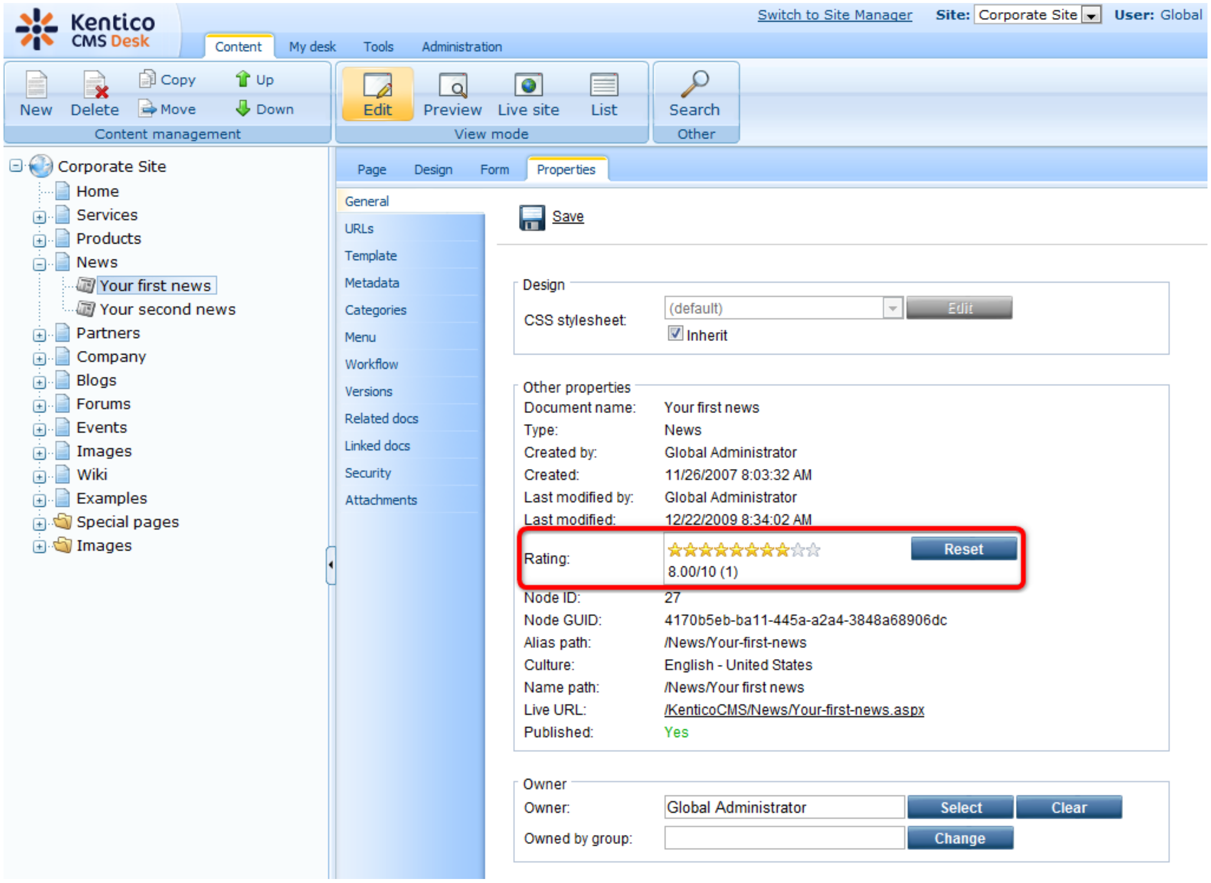
For the purposes of this example, it is also advisable to set the value of the **Show for document types** property to **cms.news**. This ensures that the web part will be displayed only for the particular news items and not for the list of news on the News section title page.

Enter the values and click **OK**.

4. Sign out of the administration interface and view the **Your first news** news item. You should see the web part below the news text as in the screenshot below. Now try rating!



5. If you switch back to **CMS Desk** and view **Properties** of the news item, you should see the **Rating** property on the **General** tab. This property reflects the current rating of the selected document. All ratings are recalculated to 10 step scale and displayed as stars here, no matter what the settings of the web part are. You can reset the value using the **Reset** button.

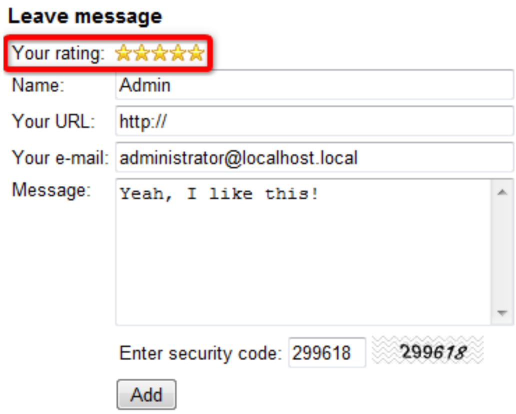


8.12.3 Other involved web parts

The **Message board** and **Group message board list** web parts enable content rating too. For both of them, you have to set the following three properties:

- **Enable content rating** - if checked, content rating will be enabled by the web part
- **Rating type** - appearance of the web part; Stars, Radio-buttons or Drop-down list can be chosen
- **Max rating value** - size of the rating scale; e.g. if 7 is entered, rating will be possible on a scale from 1 to 7

When leaving a board message, the rating control (determined by the Rating type property) will be displayed above the Name field. Users can select their rating and after sending the message, the rating will be applied.



8.12.4 Displaying ratings in transformations

Content rating results can also be displayed in transformations. To do that, the following code needs to be placed in your transformation:

```
<%@ Register Src="~/CMSAdminControls/ContentRating/RatingControl.ascx" TagName
="RatingControl" TagPrefix="cms" %>

<cms:RatingControl ID="elemRating" runat="server" Enabled="false" RatingType
="Stars" ExternalValue='<%# Convert.ToString(CMS.GlobalHelper.ValidationHelper.
GetDouble(Eval("DocumentRatingValue"), 0)/((CMS.GlobalHelper.ValidationHelper.
GetDouble(Eval("DocumentRatings"), 0) == 0?1:CMS.GlobalHelper.ValidationHelper.
GetDouble(Eval("DocumentRatings"), 1)))) %>' />
```

8.12.5 Content rating internals and API

8.12.5.1 Content rating database tables and API classes

The Content rating module uses the following database table:

- **CMS_Document -** this table is used for storing information about documents in the content tree; its *DocumentRatingValue* column stores the total sum of all ratings of a particular document, while the *DocumentRatings* column stores the total number of ratings of the document

The Content rating API is provided by the following **CMS.TreeEngine** namespace classes:

- **TreeProvider -** among other methods for document manipulation, this class provides methods for management of documents' ratings

The following topics show how methods from these classes can be used to manage abuse reports:

- Adding document rating
- Getting document rating
- Modifying document rating
- Resetting document rating

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.12.5.2 Adding document rating

The following code example explains how content rating can be added to a document using the API.

[C#]

```
using CMS.TreeEngine;
using CMS.CMSHelper;

int nodeId = 241;
double rating = 0.5f;

// Get the document
TreeProvider tree = new TreeProvider(CMSContext.CurrentUser);
CMS.TreeEngine.TreeNode node = tree.SelectSingleNode(nodeId, CMSContext.
PreferredCultureCode);

// Add rating
// If the 3rd parameter is true, information that the document has been rated is
stored in a cookie. It is possible to check if the user rated the document.
TreeProvider.AddRating(node, rating, false);
```

8.12.5.3 Getting document rating

The following code example demonstrates how you can get rating of a document using the API.

[C#]

```
using CMS.TreeEngine;
using CMS.CMSHelper;

int nodeId = 241;
double rating = 0.0f;

// Get the document
TreeProvider tree = new TreeProvider(CMSContext.CurrentUser);
CMS.TreeEngine.TreeNode node = tree.SelectSingleNode(nodeId, CMSContext.
PreferredCultureCode);

// Get its rating
rating = node.DocumentRatingValue / node.DocumentRatings;
```

8.12.5.4 Modifying document rating

The following code example shows how rating of a document can be modified using the API.

[C#]

```
using CMS.TreeEngine;
using CMS.CMSHelper;

int nodeId = 241;
double rating = 1.0f;
int numOfRatings = 2;

// Get the document
TreeProvider tree = new TreeProvider(CMSContext.CurrentUser);
CMS.TreeEngine.TreeNode node = tree.SelectSingleNode(nodeId, CMSContext.
PreferredCultureCode);

// Modify its rating
TreeProvider.SetRating(node, rating, numOfRatings);
```

8.12.5.5 Resetting document rating

The following code example shows how to reset content rating of a document using the API.

[C#]

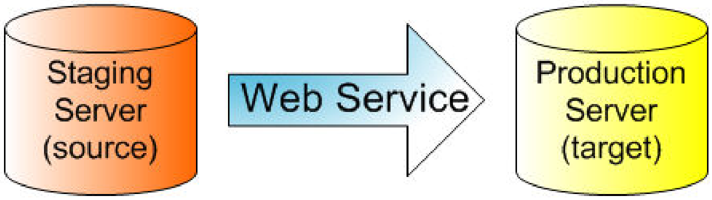
```
using CMS.TreeEngine;
using CMS.CMSHelper;

int nodeId = 241;

// Get the document
TreeProvider tree = new TreeProvider(CMSContext.CurrentUser);
CMS.TreeEngine.TreeNode node = tree.SelectSingleNode(nodeId, CMSContext.
PreferredCultureCode);

// Reset its rating
TreeProvider.ResetRating(node);
```

8.13 Content staging

8.13.1 Overview

The **Content staging** module allows you to easily transfer changes made to documents or objects in a Kentico CMS instance on one server to another instance on another server. It is also possible to perform complete synchronization of all documents and objects this way. This is particularly useful if you need to separate website content in development, editing (staging) and live (production) environment.

All documents stored in CMS Desk's content tree can be synchronized using the module. However, not all objects in a Kentico CMS system can be synchronized this way. The What can be synchronized topic gives you an overview of which data can be synchronized using the module.



For the synchronization to be possible, you need to configure Kentico CMS instances on particular servers as described in the Content staging configuration topic. Each instance can be configured as a source server or as a target server. Changes from the source server are transferred to the target servers, as depicted in the diagram above. Every site can use different target servers for content staging. You can even synchronize between different sites in the same Kentico CMS instance (in the same database).

It is also possible to have servers configured as both a target and a source server at the same time, so that changes can be transferred in both directions. This possibility and the configuration required for it is described in the Bi-directional staging topic.

Once all instances are configured, you can start synchronizing the changes. All changes are logged as staging tasks in **CMS Desk -> Tools -> Content staging**. The Synchronizing the content topic explains how staging tasks can be transferred to the other servers, as well as all other operations possible in this section of the user interface. Synchronization can also be performed automatically, based on a pre-defined scheduled task named **Content synchronization**. The Automatic synchronization topic explains this scheduled task can be enabled.

The synchronized data is transferred over a secured web service. Management of content staging tasks and performing of the actual synchronization can be allowed only to certain roles. The Security topic provides more information about permissions that need to be granted to roles in order to perform these actions.

Another way of transferring your content from one Kentico CMS instance to another is to export the data from one instance and import them to another one. The Managing sites -> Export and import chapter of this guide gives you an extensive overview of the built-in export and import functionality.

8.13.2 What can be synchronized

The Content staging module **supports** synchronization of the following data:

- Document data - documents in the content tree
- Document attachments – if a document contains document attachments or file fields, the files are synchronized together with the document
- Document relationships – if specified relationship and related document exist on the target server, the relationship is also synchronized
- Workflow process – only published document versions are synchronized to the target server and both servers need to have the same workflow schemas defined
- Custom tables and data stored in them
- Media libraries and the files and folders in them
- Global objects - all global objects except the ones listed in the list of not-supported data below

The Content staging module **does not support** synchronization of the following data:

- BizForms data, the forms themselves are synchronized
- Forum posts, the forums themselves are synchronized
- Message board messages, the boards themselves are synchronized
- Blog comments, the blogs and blog posts are synchronized
- Friends
- Messaging module messages
- ACLs (document-level permissions)
- Abuse reports
- Web analytics
- Export history
- Event log
- Web templates

Staging of Media libraries content

Kentico CMS supports staging of the **physical files and folders** stored in the file system in a Media library. Staging tasks related to Media library files are logged in CMS Desk -> Tools -> Content Staging -> Objects.

You can limit the maximal size of synchronized media library files by adding the following key to the *appSettings* section of you *web.config* file. The value is entered in **kiloBytes** and files larger than the value will not be synchronized.

```
<add key="CMSMediaFileMaxStagingSize" value="1024" />
```

Please note:

Please note

Changes in media library files and folders are logged as content staging tasks only when performed via the UI. If you make some changes directly in the library folder in the file system (e.g. upload or update some files via FTP), the changes are not logged.

Also, if you make some change to a file via the UI and then update the file via FTP, the current file (the one updated via FTP) will be transferred to the target server, even if the staging task was created before the file upload. This happens because binary data of the files are loaded in synchronization time, not when the synchronization task is logged.

8.13.3 Content staging configuration

Configuration of the Content staging module consists of the following three parts:

1. Microsoft WSE configuration - renaming of a *.dll* library on all servers
2. Source server configuration - configuration of the server from which changes will be transferred to the

target servers
3. Target server configuration - configuration of the server to which changes will be transferred from the source servers

Apart from this configuration, you also need to ensure that all instances use the same settings (document types, templates, web parts …), code files and that both servers use the same version of Kentico CMS.

Microsoft WSE configuration

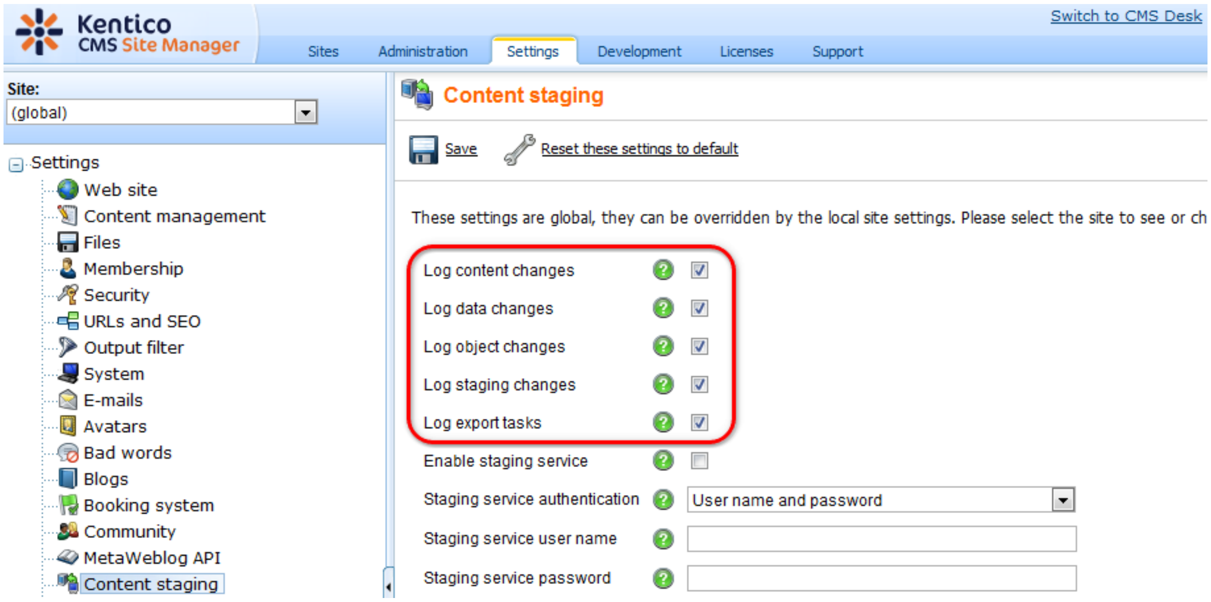
To enable Content staging, you first need to open the **bin** folder under your web project and rename the **Microsoft.Web.Services3.dll.rename** file to **Microsoft.Web.Services3.dll**. This needs to be done on both source and target servers.

Source server configuration

1. To configure a Kentico CMS instance as a source server, you first need to enable logging of content staging tasks using the following settings in **Site Manager -> Settings -> Content staging**.

- **Log content changes** - if enabled, synchronization tasks are automatically logged when content (a document) is modified
- **Log data changes** - if enabled, synchronization tasks are automatically logged when custom tables data are modified
- **Log object changes** - if enabled, synchronization tasks are automatically logged when an object is modified
- **Log staging changes** - if enabled, synchronization tasks are logged for changes made by synchronization from another server to this server; see Bi-directional staging for more details
- **Log export tasks** - if enabled, synchronization tasks are logged when an object is deleted (incremental update support)

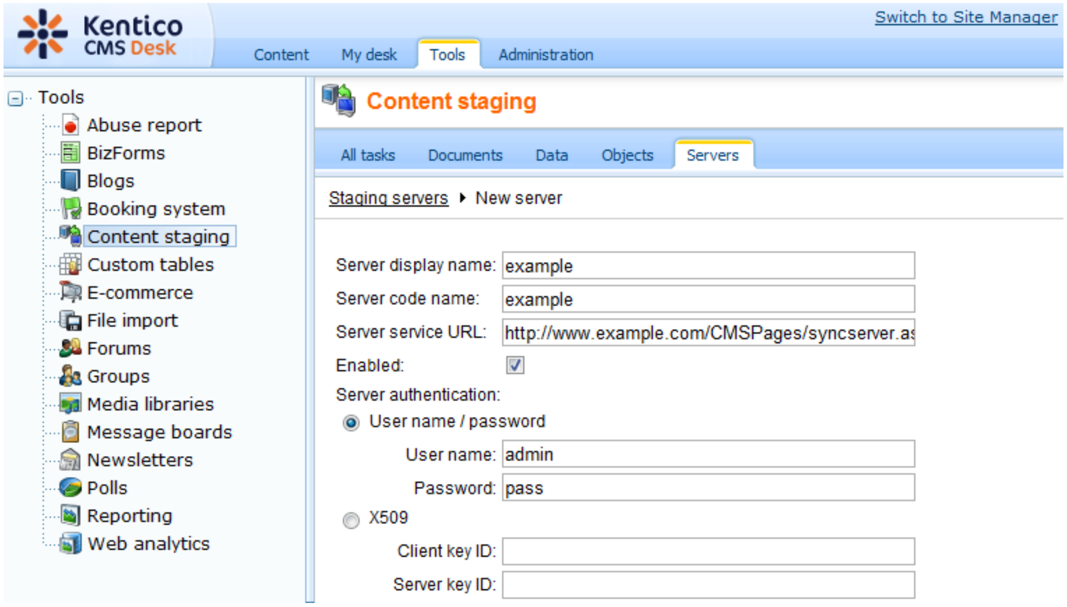
With these settings enabled, all changes to the corresponding content are logged as content staging tasks. These tasks can then be transferred to the target servers and performed there to synchronize the content.



2. Then you need to specify the target server(s). Go to **CMSDesk –> Tools –> Content staging -> Servers**.

This is where you can manage the list of target servers. To add a new server, click **New server** ( ), enter the server properties and save them:

- **Server display name** - server display name displayed to the users in the administration interface
- **Server code name** - server name used in website code
- **Server URL** - staging service URL that points to the content staging web service of the target server; the web service page is located at *~/CMSPages/syncserver.asmx*, so the URL should look like this: *http://www.example.com/CMSPages/syncserver.asmx*
- **Enabled** - if checked, the target server is enabled, it means that synchronization tasks are automatically generated for the server and that synchronization is enabled for this server; you can temporarily disable the server by un-checking the box, e.g. in case of server maintenance
- **Server authentication** - server authentication settings; you should set the same parameters that are configured for your target server (described below); the default user name is **admin** and the default password is **pass**; if you want to use X509 authentication, please consult the Using X.509 authentication topic



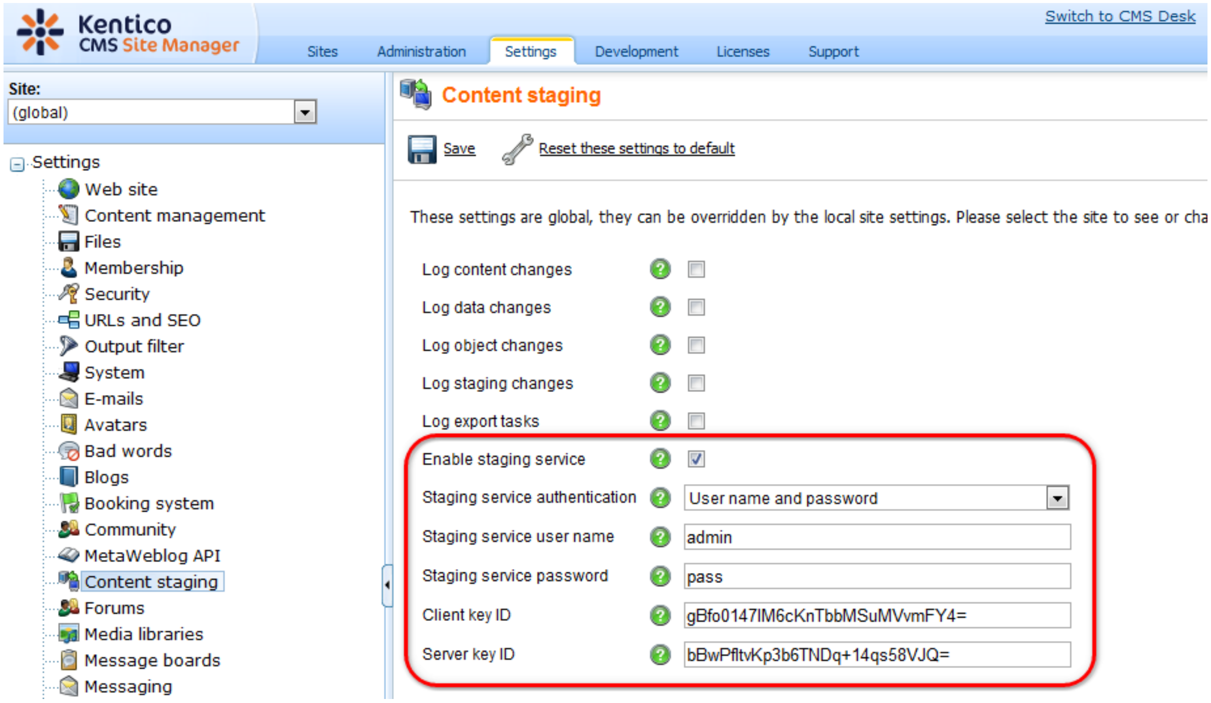
Please note

Tasks are logged only when there is at least one target server created and enabled. If there is no server created or if no server is running, no tasks are logged.

Target server configuration

On the target server, the staging service is disabled by default. For it to work, you need to set the following values in **Site Manager -> Settings -> Content staging**:

- **Enable staging service -** if checked, the staging service is enabled for the given site;
- **Staging service authentication -** staging service authentication type; it is recommended that you choose *USERNAME* authentication to configure staging first, test synchronization and then optionally configure the site for *X509* certificates
  - *USERNAME* – username/password authentication (fast, recommended for the data without high security requirements)
  - *X509* – X509 certificate authentication (more secure, slower, requires certificates); more details can be found in the Using X.509 authentication topic
- **Staging service username** and **password -** username and password for the *USERNAME* authentication
- **Server key ID** and **Client key ID -** certificate keys for the *X509* authentication



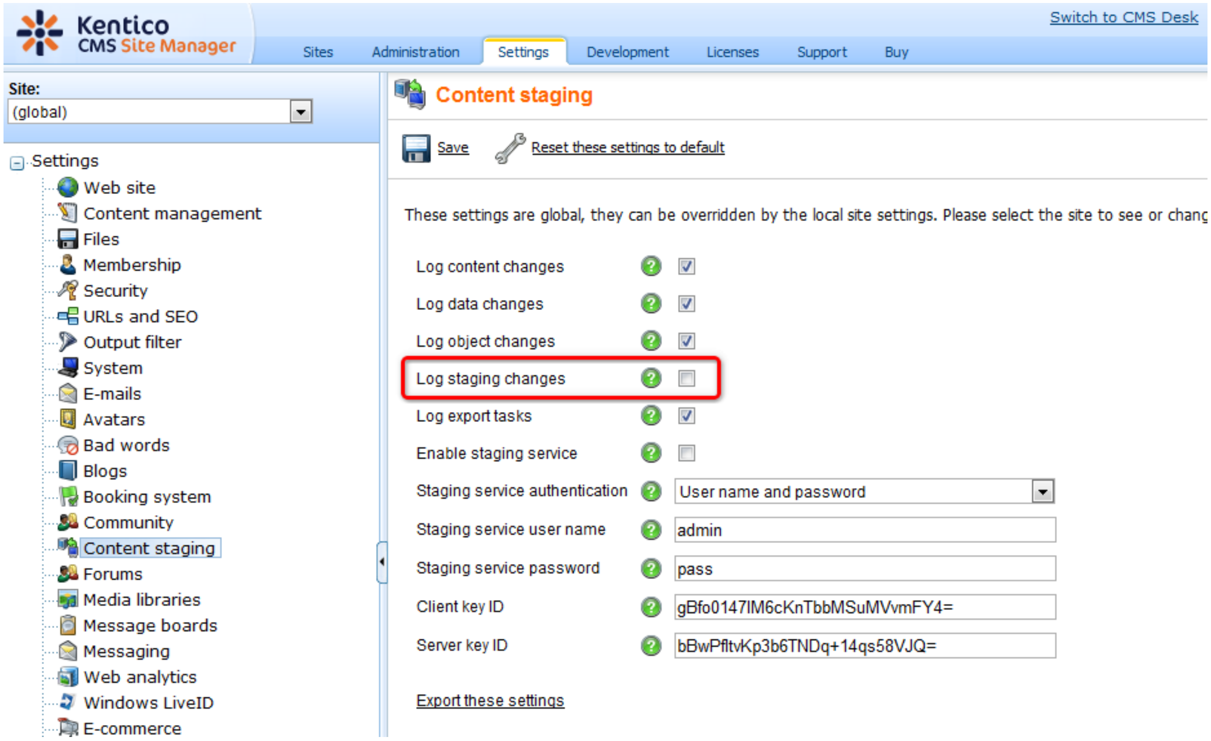
8.13.4 Bi-directional staging

If you want to perform content staging bi-directionally between **two servers**, i.e. you want to transfer changes made on Server 1 to Server 2 and also in the other direction from Server 2 to Server 1, you need to to configure both servers as a source and a target server at the same time, as described in the Content staging configuration topic.

On top of it, you also need to **disable** the **Log staging changes** option in **Site Manager -> Settings -> Content staging** on both servers.

If this option is enabled, changes made to the system via content staging synchronization (i.e. transferred from Server 1 to Server 2) are logged as new synchronization tasks on the target server (on Server 2). Now if you perform synchronization in the reversed direction, i.e. back from Server 2 to Server 1, these tasks are also performed and logged back on Server 1 as new staging tasks. This goes on and on in a never-ending loop, which results in the tasks remaining and not being deleted on both servers.

In order to prevent this behavior, you need to disable the **Log staging changes** option on both servers so that the staging changes are not logged as new synchronization tasks.



8.13.5 Synchronizing the content

All changes made to the documents and objects are tracked in the database, in the synchronization log. You can view the changes in **CMS Desk –> Tools –> Content Staging**.

The interface for viewing changes and performing synchronization is divided into the following tabs:

All tasks tab

On the **All tasks** tab, you can see a list of all content staging tasks, i.e. all changes made to the system that can be synchronized on the target server.

Using the **Server** drop-down, you can choose the target server that you want to synchronize. By choosing **(all)**, you perform synchronization for all available target servers. Then you can perform one of the following actions using the buttons at the bottom:

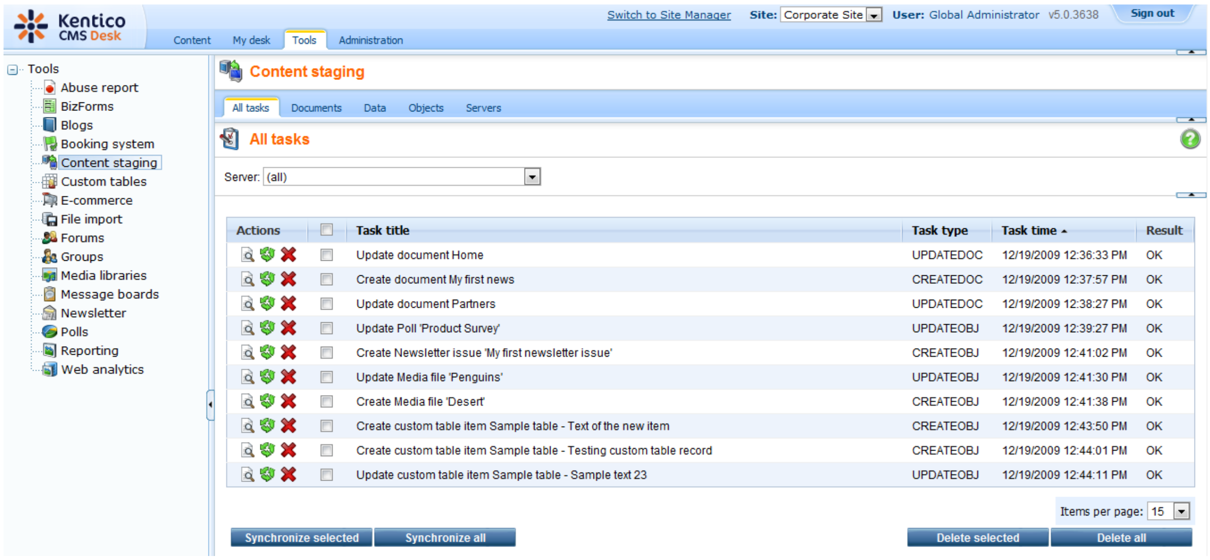
- **Synchronize selected** - performs synchronization for all tasks selected by the check-boxes (☑) on the target server;
- **Synchronize all** - performs synchronization for all listed tasks on the target server; in case you made any changes to content on the target server in the meantime, these changes will be overwritten

- **Delete selected** - deletes all tasks selected by the check-boxes (☑) on the target server
- **Delete all** - deletes all listed tasks on the target server

You can also perform the following actions separately with particular staging tasks:

- **View** - opens a new window with detailed information about the staging task
- **Synchronize** - performs the synchronization task on the target server

- **Delete** - deletes the synchronization task from the list



Documents tab

In the screenshot below, you can see the **Documents** tab. By clicking the website root in the content tree on the left, you can view a list of all changes (synchronization tasks) made to all the documents of your website. By clicking a particular document, you can view only the changes made to it.

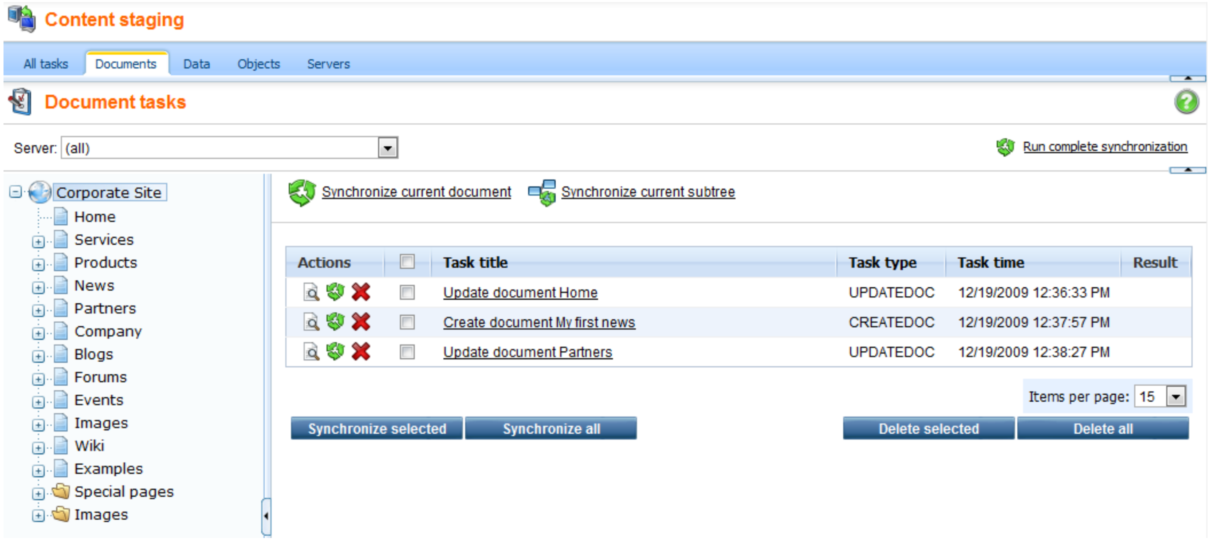
You can perform the same actions as on the **All tasks** tab, as described above.

You can also perform the following manual actions. These actions are manual because they are not related to the listed tasks and they can be performed even if there are no synchronization tasks logged:

- **Run complete synchronization** - performs complete synchronization of all documents in the content tree
- **Synchronize current document** - synchronizes the currently selected document
- **Synchronize current sub-tree** - synchronizes all documents in the selected sub-tree

The following types of tasks are logged for documents. You can see the type in the **Task type** column:

- **CREATEDOC** - document was created
- **UPDATEDOC** - document was modified
- **DELETEDOC** - document was deleted
- **DELETEALLCULTURES** - all cultural versions of the document were deleted
- **PUBLISHDOC** - document was published
- **ARCHIVEDOC** - document was archived
- **REJECTDOC** - document was rejected
- **MOVEDOC** - document was moved to another location in the content tree



Data tab

In the screenshot below, you can see the **Data** tab. This is where changes made to data in **custom tables** are logged.

You can perform the same actions as on the **All tasks** tab, as described above.

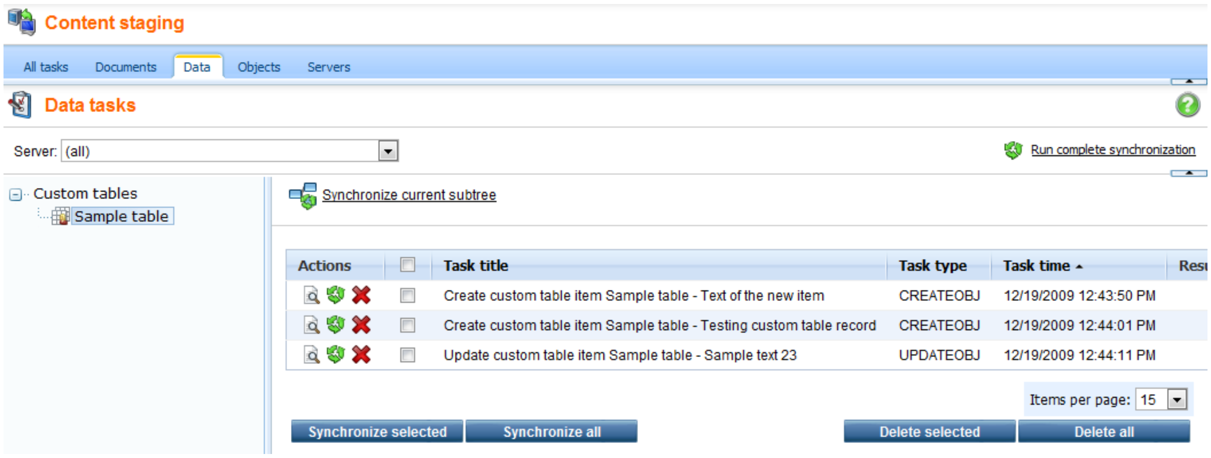
You can also perform the following manual actions. These actions are manual because they are not related to the listed tasks and they can be performed even if there are no synchronization tasks logged:

- **Run complete synchronization** - performs complete synchronization of all data in all custom tables
- **Synchronize current sub-tree -** synchronizes all data in the selected custom table

The following types of tasks are logged for custom tables data. You can see the type in the **Task type** column:

- **CREATEOBJ** - new item was added to a table
- **UPDATEOBJ** - an item in a table was updated
- **DELETEOBJ** - an item in a table was deleted

Please note: Staging tasks for custom table items are logged based on their *ItemGUID* column. Items which do not have this column (typically in custom tables imported from older versions of Kentico CMS), changes made to them are not logged. However, you can edit the custom table in CMS Desk -> Tools -> Custom tables. You will see a warning message with a link letting you to generate the GUIDs for the table.



Objects tab

In the following screenshot, you can see the **Objects** tab. This is where changes made to objects are logged.

The main categories are **website** and **Global objects**. The first category contains object changes connected to the current website, whilst the second one contains object changes for global objects.

You can perform the same actions as on the **All tasks** tab, as described above.

When you select a particular object category, you get also the following manual action offered. This action is manual because it is not related to the listed tasks and can be performed even if there are no synchronization tasks logged:

- **Synchronize current sub-tree -** synchronizes all objects in the selected category

The following types of tasks are logged for objects. You can see the type in the **Task type** column:

- **CREATEOBJ** - new object was created
- **UPDATEOBJ** - object was modified
- **DELETEOBJ** - object was deleted
- **ADDTOSITE** - object was assigned to a site; applicable only to site-related objects
- **REMOVEFROMSITE** - object was removed from a site; applicable only to site-related objects

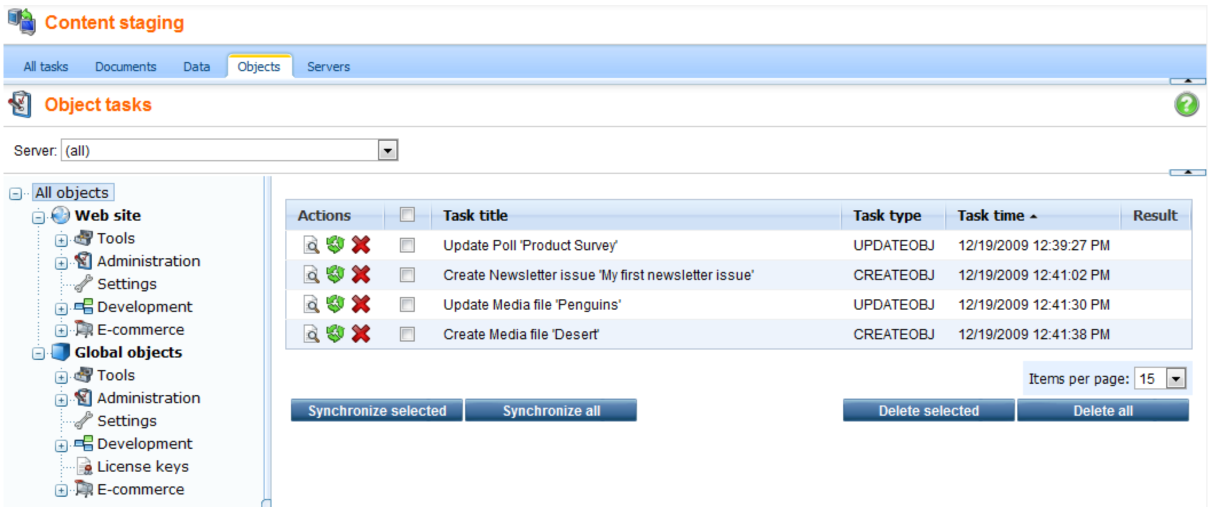
The following types of tasks are logged for folders in media libraries:

- **CREATEFOLDER** - folder was created
- **RENAMEFOLDER** - folder was renamed
- **COPYFOLDER** - folder was copied
- **MOVEFOLDER** - folder was moved
- **DELETEFOLDER** - folder was deleted

Please note

Global metadata changes such as changes to document types, custom tables and system tables produce staging tasks for all staging servers of all sites. In such case, it is recommended to synchronize such changes to all servers of all sites at the same time to prevent overwriting of such metadata and loosing the data by synchronizing the older tasks later.

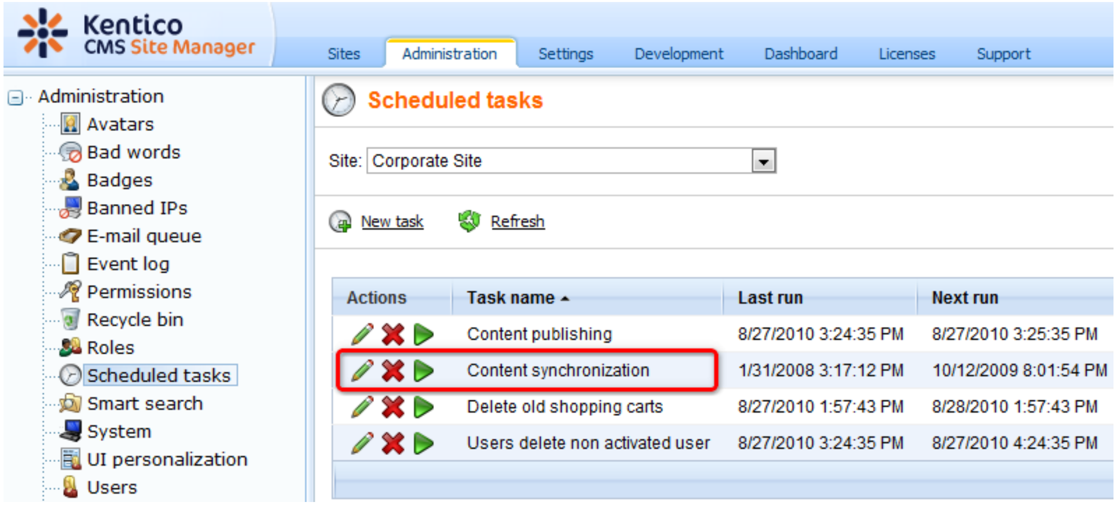
You can use the *<add key="CMSStagingTreatServerNamesAsInstances" value="true" />* key to make sure that once the global task is synchronized, it is deleted from all other servers with the same name to prevent such possibility. The default value is false since staging can use multiple target instances targeted with the same names.



8.13.6 Automatic synchronization

Automatic synchronization is available **only for document changes**. Object changes cannot be synchronized this way. Manual synchronization remains the only way to perform object synchronization tasks.

If you want the changes to be synchronized from the staging to the live site on a regular basis without waiting for the administrator's approval, you can configure the scheduled task called **Content synchronization** in **Site Manager -> Administration -> Scheduled tasks**. This task is disabled by default, so you need to enable it and configure the synchronization interval.



More information about the built-in scheduler in Kentico CMS can be found in the Development -> Scheduler chapter of this guide.

8.13.7 Using X.509 authentication

In order to use **X.509** authentication, you need to install your own certificates, or you may use our sample ones.

Using the sample certificates

Kentico CMS is delivered with sample client and server private certificates. In order to install them, you need to do the following on the source server and on the target server:

1. Choose **Start -> Run**, type **mmc** and press **Enter**.

2. In the console window, choose **File -> Add/Remove Snap-in**.

3. Click **Add** and choose **Certificates**.

4. Choose **Computer account** in the next step.

5. Choose **Local computer** in the next step. Finish adding the **Certificates** snap-in.

6. Unfold **Certificates (Local Computer)** under the console root, right-click **Personal** and choose **All Tasks -> Import...** The **Certificate Import Wizard** starts.

7. Import the appropriate certificate from the appropriate **.pfx** file in **C:\Program Files\Kentico CMS\<version>\SampleCertificates**. Use *Client private.pfx* for the client certificate and *Server private.pfx* for the server certificate.

8. Enter the following password for the sample certificates (it is the same for the client and the server certificate): **wse2qs**

9. Now you need to grant the **READ** permissions for the certificate file to the ASP.NET account (names of the account under different operating systems are described in the Disk permissions problems

chapter). You can do that using the **WseCertificate3.exe** tool that can also be found in **C:\Program Files\Kentico CMS\<version>\Sample Certificates**

10. Run the **WseCertificate3.exe** tool.

11. Choose **Local Computer** in the **Certificate Location** field.

12. Choose **Personal** in the **Store Name** field.

13. Click **Open Certificate** and choose either the client or the server certificate.

14. Click **View Private Key File Properties...** and grant the **READ** permission for this file to the ASP. NET account (names of the account under different operating systems are described in the Disk permissions problems chapter).

Using your own certificates

If you're using your own certificates (highly recommended), you will need to update the following values in the **Site Manager -> Settings -> Content Staging**:

- Client key ID
- Server key ID

To get these IDs, you can use the **WseCertificate3.exe** tool located in **C:\Program files\KenticoCMS\<version>\SampleCertificates**.

1. Run the **WseCertificate3.exe** tool.

2. Choose **Local Computer** in the **Certificate Location** field.

3. Choose **Personal** in the **Store Name** field.

4. Click **Open certificate** and select either client or the server certificate. In the **Key identifiers group** you can now see the certificate key, **Windows key identifier (Base64)** should be used within Kentico CMS settings.

Important: Sample certificates

Using the sample certificates is not secure and it's also very slow. It's highly recommended that you use your own certificate issued by a certification authority.

Tip

If you encounter problems with content staging when using SLL (X.509), you may try adding the following key to your *web.config* file:

```
<add key="CMSStagingAcceptAllCertificates" value="true" />
```

This key ensures that all certificates will be accepted. If set to false, only certificates issued by a certification authority will be accepted.

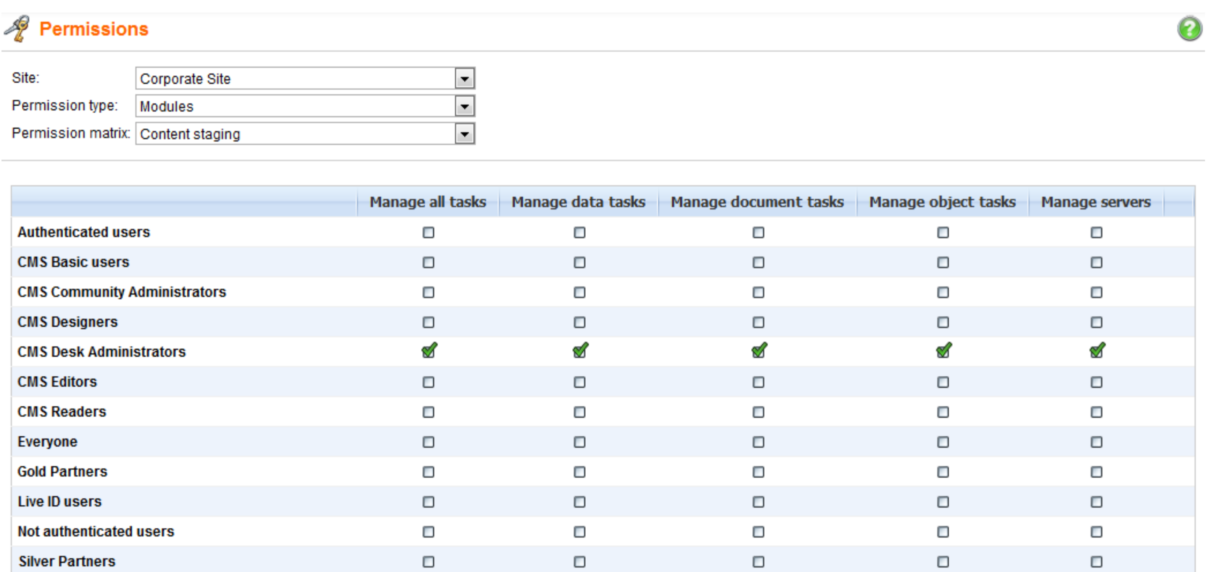
8.13.8 Security

You can control the access to the Content staging module in **Administration -> Permissions**. You need to select:

- **Site**: your site
- **Permission type**: Modules
- **Permission matrix**: Content staging

In the permission matrix, you can grant the following permissions to the particular roles:

- **Manage all tasks** - displays the **All tasks** tab and allows synchronization and management of synchronization tasks on it
- **Manage data tasks** - displays the **Data** tab and allows synchronization and management of synchronization tasks on it
- **Manage document tasks** - displays the **Documents** tab and allows synchronization and management of synchronization tasks on it
- **Manage object tasks** - displays the **Objects** tab and allows synchronization and management of synchronization tasks on it
- **Manage servers** - allows members of the roles to manage target server configurations on the **Servers** tab



	Manage all tasks	Manage data tasks	Manage document tasks	Manage object tasks	Manage servers
Authenticated users	☐	☐	☐	☐	☐
CMS Basic users	☐	☐	☐	☐	☐
CMS Community Administrators	☐	☐	☐	☐	☐
CMS Designers	☐	☐	☐	☐	☐
CMS Desk Administrators	☑	☑	☑	☑	☑
CMS Editors	☐	☐	☐	☐	☐
CMS Readers	☐	☐	☐	☐	☐
Everyone	☐	☐	☐	☐	☐
Gold Partners	☐	☐	☐	☐	☐
Live ID users	☐	☐	☐	☐	☐
Not authenticated users	☐	☐	☐	☐	☐
Silver Partners	☐	☐	☐	☐	☐

8.13.9 Synchronization using API

In special cases, you may need to use Kentico CMS API to perform your own synchronization process. There are several methods you can use to work with synchronization.

WorkflowEngine.DocumentHelper. LogSynchronization	Creates a synchronization task for the specified document. This method does not actually perform the synchronization, it only creates the task which can be later synchronized. By default, the method creates the task for all the enabled servers. You can use the overriden version with serverId parameter to specify the particular server.
CMSStaging.StagingHelper. RunSynchronization	Runs the synchronization of specified task for the specified server or servers.

Here is an example code how to synchronize the content of the document "/Home" to server "Staging.Target1":

[C#]

```
using CMS.TreeEngine;
using CMS.CMSHelper;
using CMS.Staging;
using CMS.WorkflowEngine;

TreeProvider tree = new TreeProvider(CMSContext.CurrentUser);
// Get the base document record
TreeNode node = tree.SelectSingleNode(CMSContext.CurrentSiteName, "/Home", "en-us"
, false, null, false);
if (node != null)
{
  // Get full document record (required for synchronization logging)
  node = tree.SelectSingleNode(CMSContext.CurrentSiteName, node.NodeAliasPath,
node.DocumentCulture, false, node.NodeClassName, false);
  if (node != null)
  {
    // Get the server
    ServerInfo si = ServerInfoProvider.GetServerInfo("Staging.Target1",
CMSContext.CurrentSiteID);
    if (si != null)
    {
      // Log the synchronization task for the given server
      TaskInfo ti = DocumentHelper.LogSynchronization(node, TaskTypeEnum.
UpdateDocument, tree, si.ServerID);
      if (ti != null)
      {
        // Run the task synchronization
        StagingHelper.RunSynchronization(ti.TaskID, si.ServerID);
      }
    }
  }
}
```

You can also use low level operations from TaskInfoProvider, SynchronizationInfoProvider and ServerInfoProvider to achieve synchronization, but previous methods should be enough to perform any simple synchronization actions.

The following example shows how you can synchronize the administrator user account. Synchronization of any other objects is done the same way using the API, as all objects implement the **IInfoObject** interface.

[C#]

```
using CMS.SiteProvider;
using CMS.Staging;
using CMS.CMSHelper;

// Gets the object
 UserInfo userObj = UserInfoProvider.GetUserInfo("administrator");
 if (userObj != null)
 {
  // Gets the server
   ServerInfo si = ServerInfoProvider.GetServerInfo("Staging.Target1", CMSContext.
CurrentSiteID);
   if (si != null)
   {
    // Creates the synchronization task
     TaskInfo ti = TaskInfoProvider.LogSynchronization(userObj, TaskTypeEnum.
UpdateObject, null, null, si.ServerID);
     if (ti != null)
     {
      // Runs the synchronization task
       StagingHelper.RunSynchronization(ti.TaskID, si.ServerID);
     }
   }
 }
```

8.14 Custom tables

8.14.1 Overview

The Custom tables module allows users to create their own tables in the system database and manage data in them via Kentico CMS user interface, without the need to use Microsoft SQL Server Management Studio or any other database management tool. This may come in handy in many scenarios, typically when you need to store a large number of structured data items, for which standard Kentico CMS documents are not efficient. Another typical scenario may be development of your custom modules, as you can store module data in custom tables and access them conveniently from your code using the provided API.

User interface of the module can be found at two locations, while each of them is intended for different purposes. Custom tables themselves can be created and managed in **Site Manager -> Development -> Custom tables**. This section is intentionally located in Site Manager, so that only global administrators can create the tables. The Creating custom tables topic describes creation of custom tables by means of the **New custom table wizard**, which is accessible from this part of the user interface. Once a custom table is created, its configuration (not the actual data stored in it) can be managed in the same section of the user interface, as described in the Managing custom tables topic.

The other user interface related to this module can be found in **CMS Desk -> Tools -> Custom tables**.

This is where new data records can be inserted into the tables and where existing data records can be viewed, modified or deleted. It is located in CMS Desk, so that manipulation with data in custom tables is available to content editors with appropriate permissions, as described in the Security topic. Management of the actual data items is described in Managing data in custom tables.

Actions	Item ID	Item created by	Item created when	Item modified by	Item modified when	Item order ▴	Item text
	6	53	10/20/2008 1:42:52 PM	53	10/7/2009 2:54:26 PM	1	Sample text 2
	7	53	10/20/2008 1:43:01 PM	53	10/7/2009 2:54:34 PM	2	Sample text 3
	5	53	10/20/2008 1:42:40 PM	53	12/18/2009 9:59:18 AM	3	Sample text 1
	8	53	3/5/2010 6:51:55 PM			4	Sample text 4

Items per page: 25

The module comes with three web parts that can display data from custom tables on the live site. They are described in the Available web parts topic. Use of transformations is essential with these web parts, so we have prepared the Transformations for custom tables topic. There, you can find information on how to modify the default transformations in order to change the way the data items are displayed.

Finally, the module comes with an API, which enables you to handle custom table data in your custom code. Code examples, along with a list of database tables and API classes related to the module, can be found in the Custom tables internals and API sub-chapter.

The Custom tables module interacts with two other Kentico CMS modules. Data stored in the tables can be searched using the Smart search module. You can also create Alternative forms for the tables. These forms can be used instead of the default forms for creating or editing a custom table in the administration interface, as described in Modules -> Alternative forms -> Automatically used alternative forms.

8.14.2 Creating custom tables

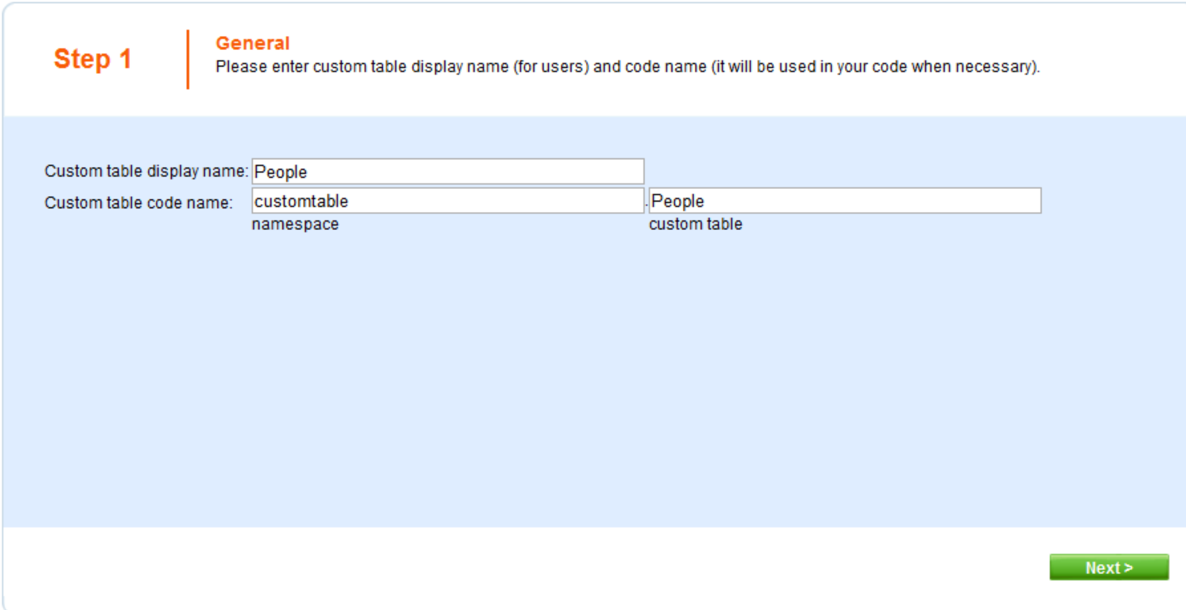
In the following example, we will create a new custom table using the **New custom table wizard**. It will be a simple table for storing information about people. Each data record in the table will contain a person's first name, last name and date of birth.

1. Go to **Site Manager -> Development -> Custom tables** and click the **New custom table** () link.

2. The **New custom table wizard** starts. In the first step, you are asked to fill in the custom table's display name (used in Kentico CMS user interface) and code name (used in website code). The code name is always preceded by a namespace, which allows you to have different tables of the same name used in different contexts. Enter the following details:

- **Custom table display name:** People
- **Custom table code name:** customtable.People

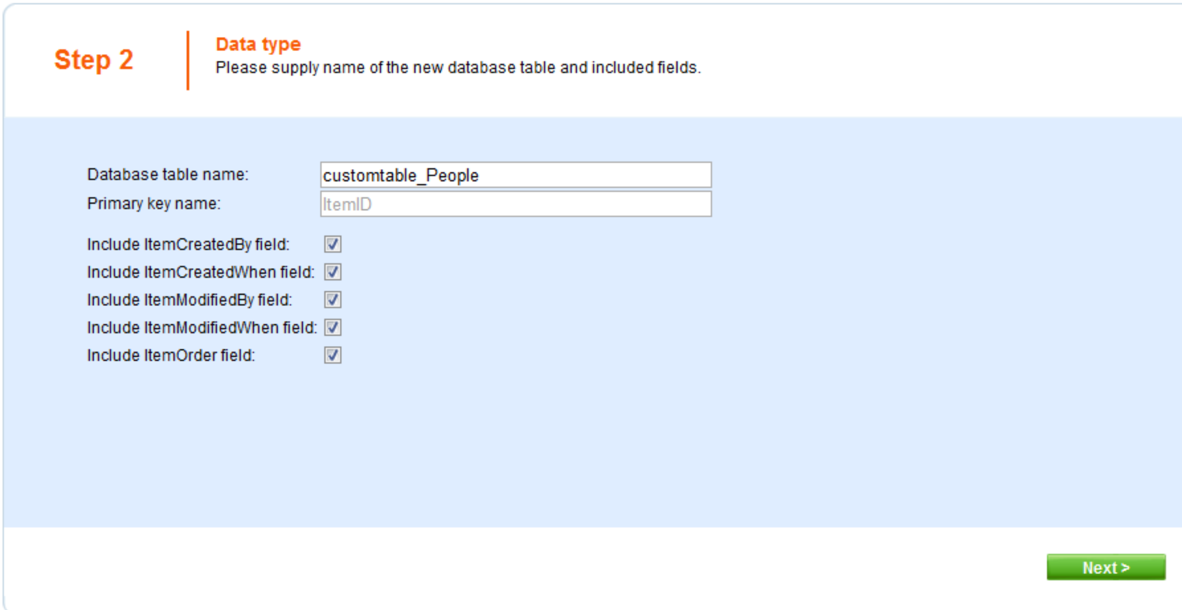
and click **Next** to continue.



3. In the second step, you are asked to fill in the database table name, i.e. the actual name of the table in the system database. A name in the <*namespace*>_<*code name*> format is pre-filled automatically. The primary key name can not be changed by default. Using the check-boxes below, you can determine which of the default fields should be included in the table:

- **ItemCreatedBy** - user name of the user who created the item
- **ItemCreatedWhen** - date and time of when the item was created
- **ItemModifiedBy** - user name of the user who last modified the item
- **ItemModifiedWhen** - date and time of last modification
- **ItemOrder** - order of the item when table content list is displayed; the lower number, the earlier position in the list

For the purposes of this example, leave default values for all options and click **Next** to continue.



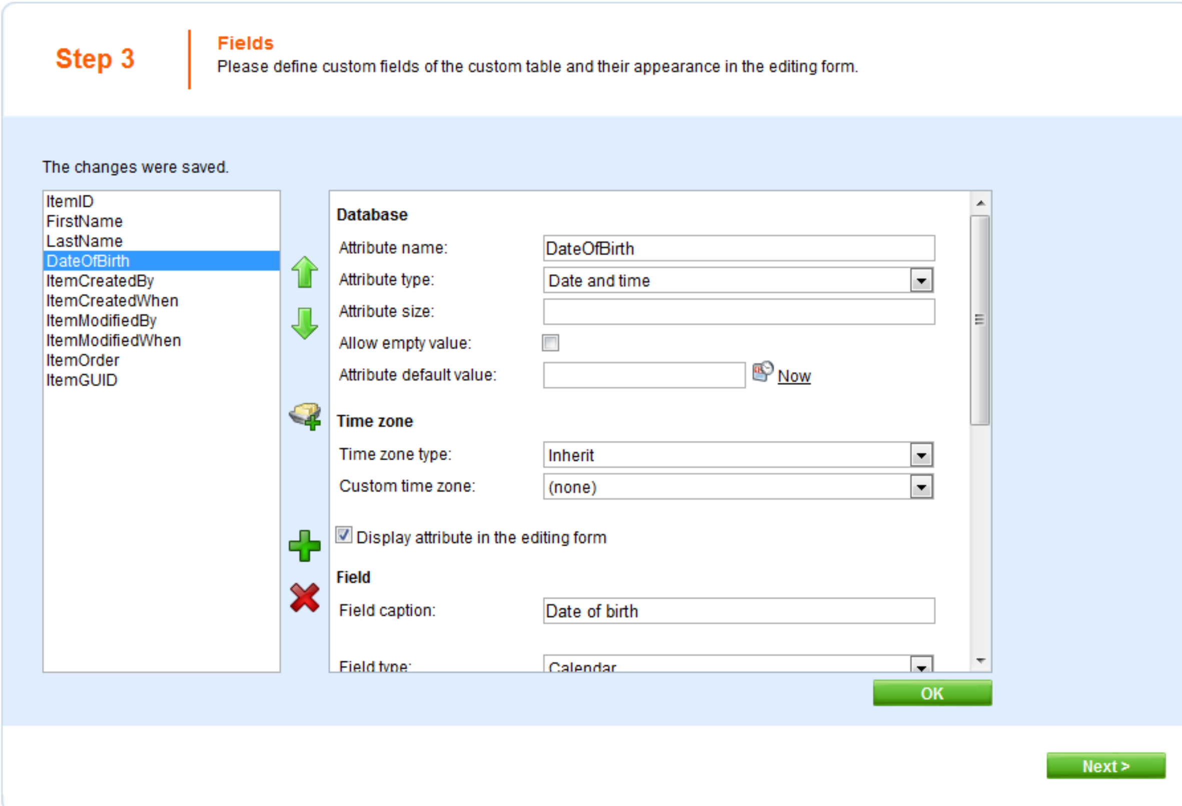
4. In the third step, the field editor is displayed. It lets you define which columns will be included in the database table. Create three fields with the following attributes:

- **Attribute name:** FirstName
- **Attribute type:** Text
- **Attribute size:** 100
- **Field caption:** First name
- **Field type:** Text box

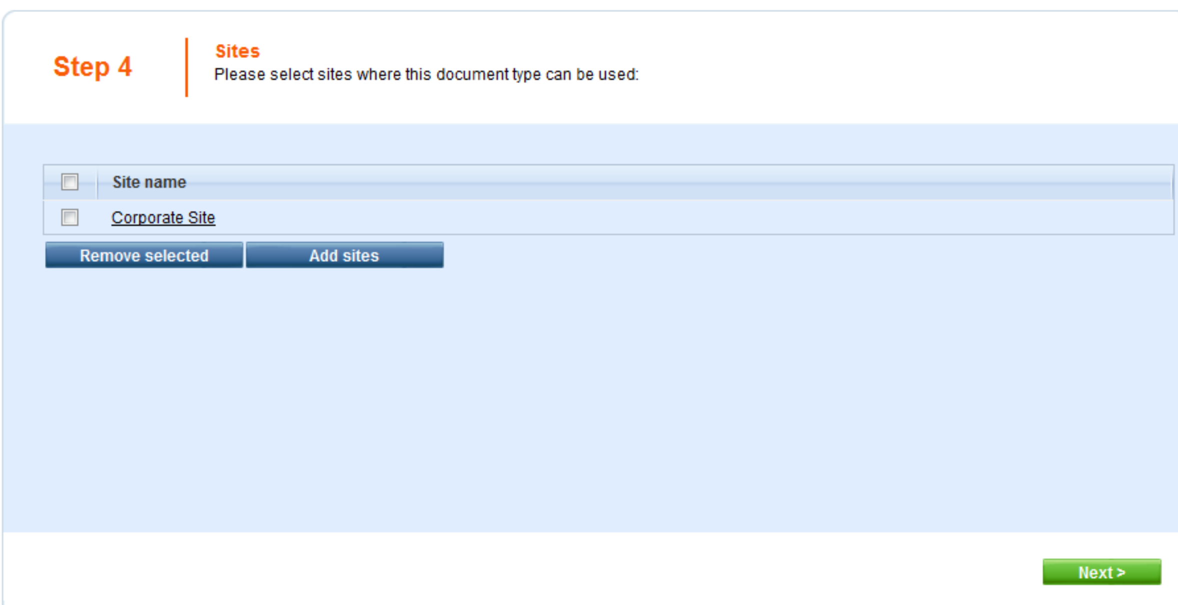
- **Attribute name:** LastName
- **Attribute type:** Text
- **Attribute size:** 100
- **Field caption:** Last name
- **Field type:** Text box

- **Attribute name:** DateOfBirth
- **Attribute type:** Date and Time
- **Field caption:** Date of birth
- **Field type:** Calendar
- **Edit time:** unchecked

When you are finished, click **Next** to proceed to the following step.



5. In Step 4, you only have to select in which sites will the custom table be available. Click **Add sites** and choose **Corporate Site** in the pop-up dialog. Finally click **Next** to proceed to the following step.

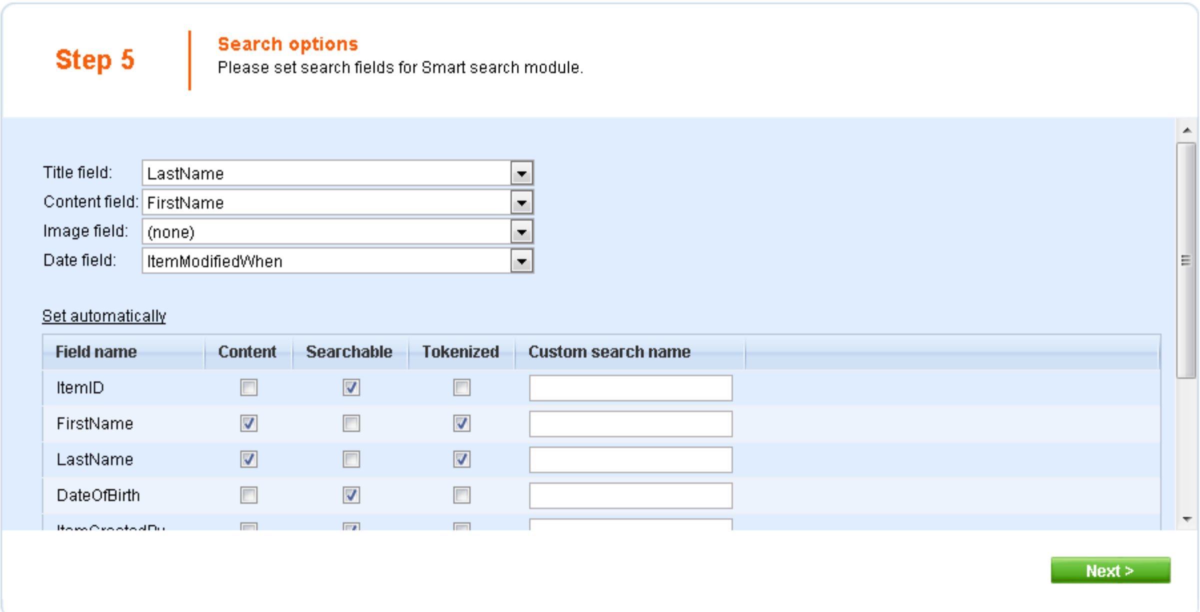


6. Step 5 offers you to configure indexing of data in the custom table by the Smart Search module. The four drop-downs at the top determine which fields of the table will be displayed in search results. Change

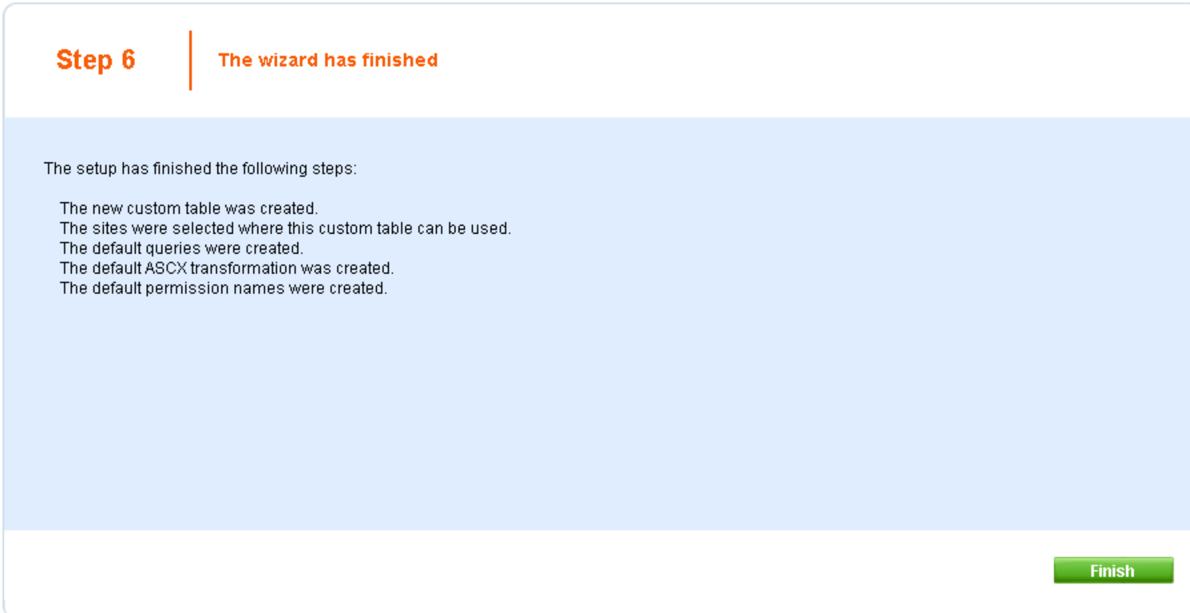
the values to the following:

- **Title field**: LastName
- **Content field**: FirstName
- **Image field**: none
- **Date field**: ItemModifiedWhen

Leave the rest of the settings at default values and click **Next**. For more information about settings in this step, please refer to Modules -> Smart search -> Object settings.



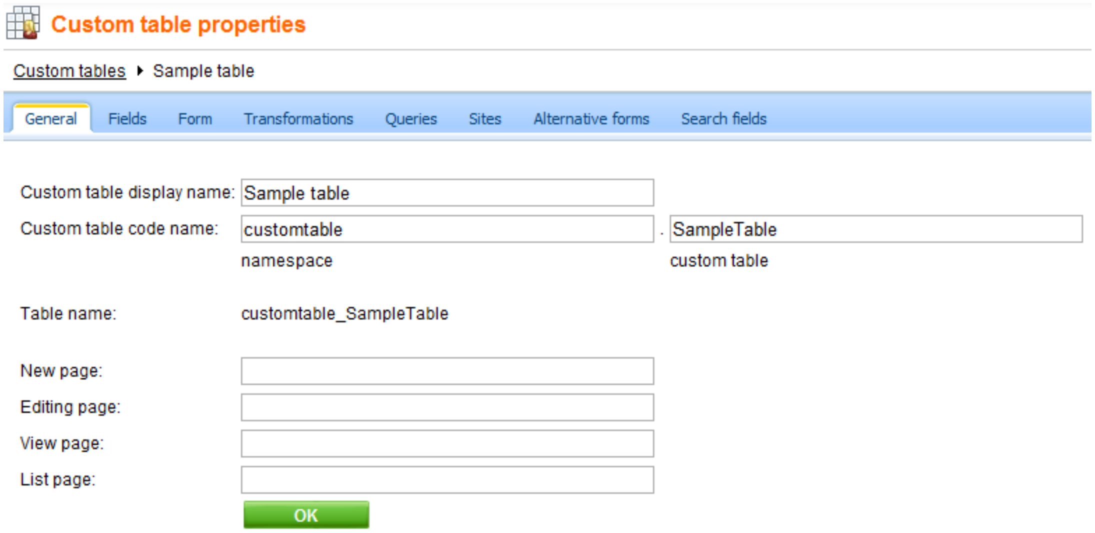
7. The fifth step gives you an overview of what has been done automatically by the wizard. To finish the wizard, click **Finish**. You will be redirected back to the list of custom tables in **Site Manager -> Development -> Custom tables**.



8.14.3 Managing custom tables

The administration interface for management of custom tables is located in **Site Manager -> Development -> Custom tables**.

In this section, you can create new custom tables as described in the [Creating custom tables]() topic. You can also **Edit** () or **Delete** () existing tables in the list. If you choose to edit () an existing custom table, you will be displayed with the user interface depicted in the screenshot below.



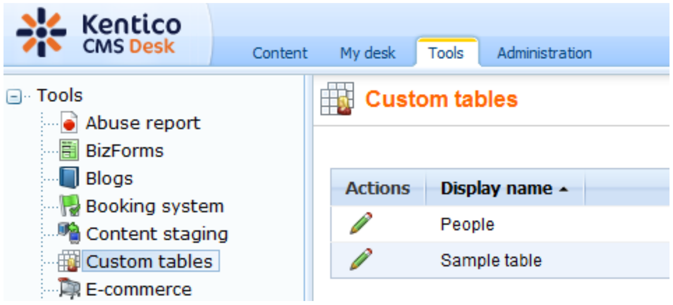
As you can see, the interface is divided into the following tabs:

- **General** - this is where you can modify the display and code name of the custom table, as well as define URLs of custom pages that can be used instead of the default pages for adding, editing, viewing and listing of data items stored in the table
- **Fields** - on this tab, you can find the field editor that can be used to manage fields (columns) of the table
- **Form** - this tab allows you to create a custom form layout that will be used for adding and editing data items
- **Transformations** - this is where transformations for the custom table can be created and managed
- **Queries** - database queries for manipulation with data in the table can be created and managed on this tab
- **Sites** - on this tab, you can define websites for which the custom table will be available
- **Alternative forms** - alternative forms for adding and editing custom table data can be created on this tab; more info can be found in Modules -> Alternative forms -> Automatically used alternative forms
- **Search fields** - on this tab, you can define how data stored in the custom table will be indexed by the Smart search module

More information on the particular tabs can be found in **Kentico CMS Context Help**, which is accessible by clicking the icon in the top right corner of each tab.

8.14.4 Managing data in custom tables

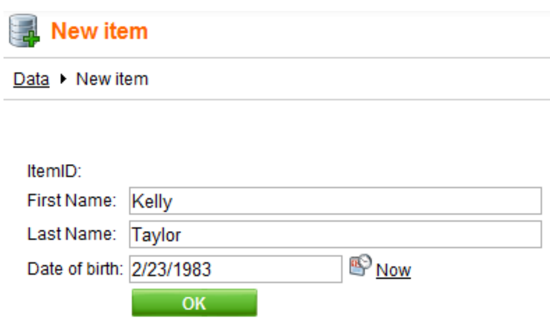
The administration interface for management of data items stored in custom tables is located in **CMS Desk -> Tools -> Custom tables**.



In the following example, you will learn how to add data into the sample **People** custom table we have created in the [Creating custom tables]() chapter. The procedure is the same for any other custom table (except for the fields, which are always specific for a particular table):

1. Click the **Edit** ( ) icon next to the **People** table. A blank page with a header as in the following screenshot will be displayed. The page is blank because there is no data in the custom table yet.

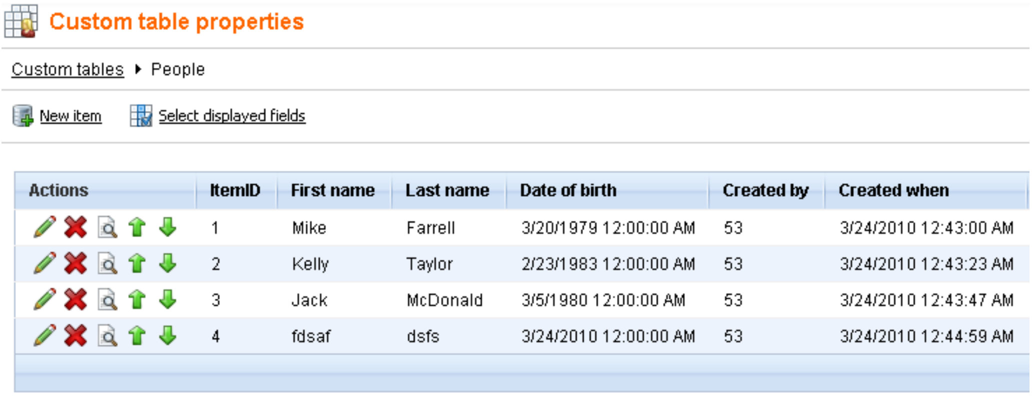
2. Click the **New item** link. The form that you can see in the following screenshot will be displayed. Enter some testing data and click **OK**.



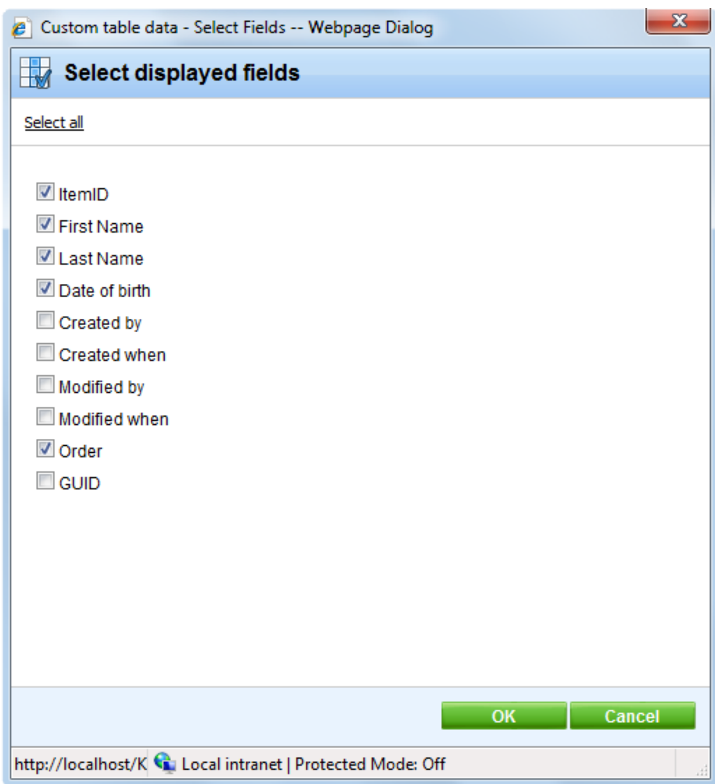
3. The data you entered has been saved into the custom table. Now you can either edit it or use the **Create another** link to create another data record. Try creating at least two other records.

4. When you're finished, switch back to the **Custom tables** listing in **CMS Desk -> Tools** and choose to **Edit** ( ) the **People** custom table. You should see a list of entries where the blank page was in step 1, before entering the records.

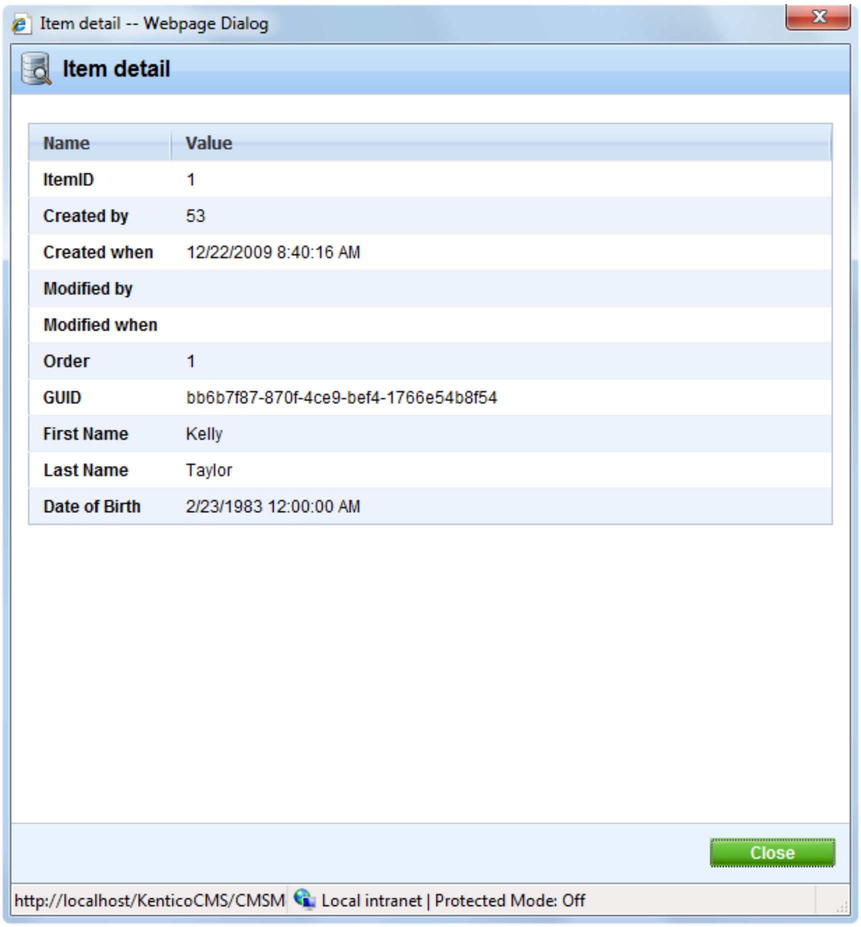
The data records can be **Edited** ( ) or **Deleted** ( ). You can also re-order records in the listing using the **Up** ( ) and **Down** ( ). The order is be stored in the **ItemOrder** property of each data record.



5. You can choose which fields should be displayed in the listing by clicking the **Select displayed fields** link. The window depicted in the following screenshot will be displayed. Using the check-boxes, you can determine which fields are to be displayed in the listing. When you make the selection and click **OK**, you should see the result immediately.



6. If you click the **View** ( ) icon next to any of the data records, detailed information about the record will be displayed in a pop-up window. All details will be displayed, no matter which fields are selected to be displayed in the listing.



8.14.5 Available web parts

The Custom tables module comes with three web parts that can be used for displaying custom table data. This page gives just a brief overview of these web parts. Detailed description of each web part and its properties can be found in Kentico CMS Web Parts Reference. Live site examples of these web parts can be found on the sample Corporate Site, under **Examples -> Webparts -> Custom tables**.

Custom table datagrid

This web part displays a grid with selected columns of data records from a custom table. Design of the web part can be modified using standard ASP.NET skins. If you want to handle the data by transformations, use one of the web parts below.

FirstName ▲	LastName	DateOfBirth
Jane	Doe	6/8/1970 12:00:00 AM
Kelly	Taylor	2/23/1983 12:00:00 AM
Mike	Farrel	12/18/1973 12:00:00 AM

Custom table datalist

In the screenshot below, you can see the basic appearance of the Custom table datalist web part. The appearance can be changed using transformations. You can find more information about using transformations with custom tables web parts in the Transformations for custom tables topic.

First name:	Kelly	**First name:**	Mike
Last name:	Taylor	**Last name:**	Farrel
Date of birth:	2/23/1983 12:00:00 AM	**Date of birth:**	12/18/1973 12:00:00 AM

Custom table repeater

In the screenshot below, you can see the basic appearance of the Custom table repeater web part. The appearance can be changed using transformations. You can find more information about using transformations with custom tables web parts in the Transformations for custom tables topic.

First name:	Kelly
Last name:	Taylor
Date of birth:	2/23/1983 12:00:00 AM

First name:	Mike
Last name:	Farrel
Date of birth:	12/18/1973 12:00:00 AM

First name:	Jane
Last name:	Doe
Date of birth:	6/8/1970 12:00:00 AM

8.14.6 Transformations for custom tables

Use of transformations is essential for web parts of the custom tables module. In the following example, you will learn how to modify the way data is displayed by modifying default transformation code.

We will use the **Examples -> Webparts -> Custom tables -> Custom table repeater** example on the sample Corporate Site. We will also use the **People** custom table that we have created in the Creating custom tables topic and populated with some data items in the Managing data in custom tables topic.

1. Go to **CMS Desk**, switch to **Edit** mode and select **Examples -> Webparts -> Custom tables -> Custom table repeater**. By default, the repeater is bound to the **Sample table**, which is a sample custom table contained in the Corporate site. It should look as in the following screenshot:

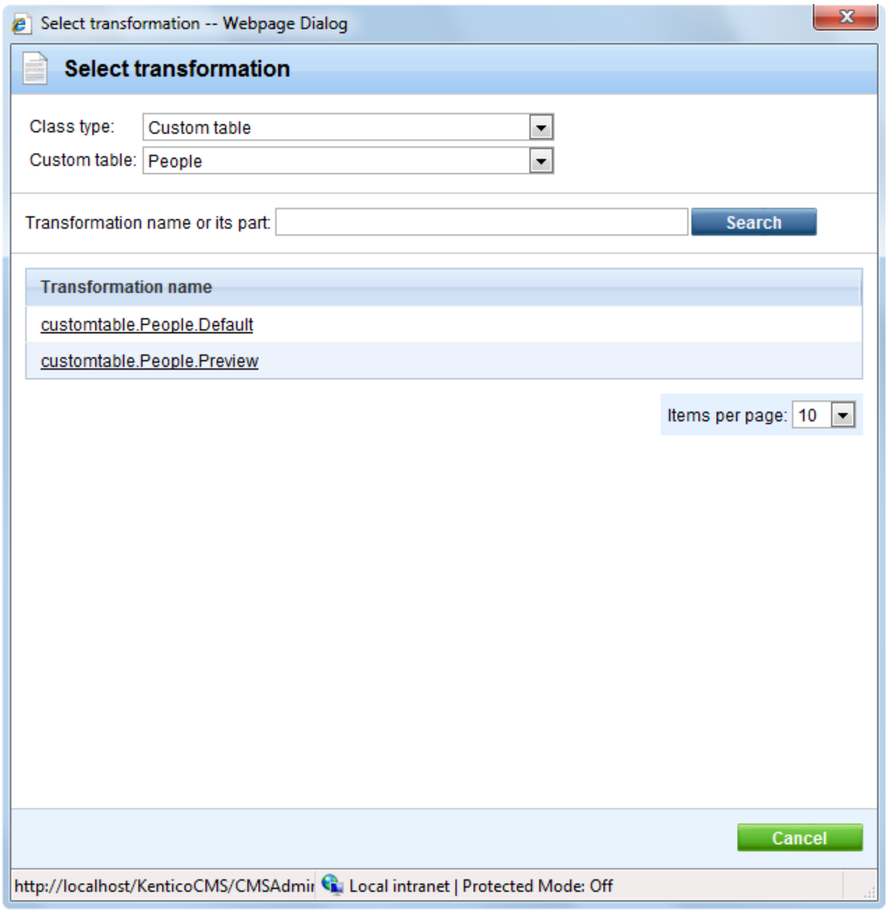
Examples > Webparts > Custom tables > Custom table repeater

Item text: Sample text 1
Item text: Sample text 2
Item text: Sample text 3

2. Switch to the **Design** tab and choose to **Configure** () the repeater's properties. First, use the **Custom table** drop-down list to select the custom table from which the data will be displayed - it should be the **People** table in our case.

3. Now it's time to choose a transformation via the **Transformation name** property. If you click the **Select** button next to the mentioned field, you will get the pop-up window as depicted in the following screenshot. There are two default transformations generated by default for each custom table - **Default**

and **Preview**. Choose the **Default** transformation and click **OK**. Click **OK** again to close the **Web part properties** window.



4. Now if you switch to the **Live site** mode, you should see that the repeater is displaying something like the following:

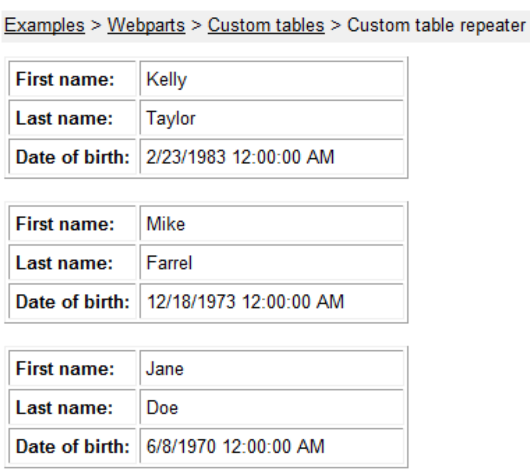
5. As you probably wouldn't want to have all the empty fields displayed and the table to be unformatted on your real website, you will need to edit the transformation code. To do it, switch to the **Edit** mode again, switch to the **Design** tab, choose to **Configure** () the web part's properties and click the **Edit**

button next to the **Transformation name** property. Now you can edit the transformation code according to your specific needs.

For the purpose of this tutorial, please replace the transformation code with the following code example. You can notice that it is just a modification of the former **Default** transformation, with the unwanted fields deleted and the highlighted tags and properties added:

```
<table border="1" cellpadding="4">
<tr>
<td><b>First name:</b></td>
<td width="180"><%# Eval("FirstName") %></td>
</tr>
<tr>
<td><b>Last name:</b></td>
<td><%# Eval("LastName") %></td>
</tr>
<tr>
<td><b>Date of birth:</b></td>
<td><%# Eval("DateOfBirth") %></td>
</tr>
</table>
<br/>
```

6. Click **Save**. Now if you go to the live site, you should see that the data is displayed in a formatted way, as depicted below.



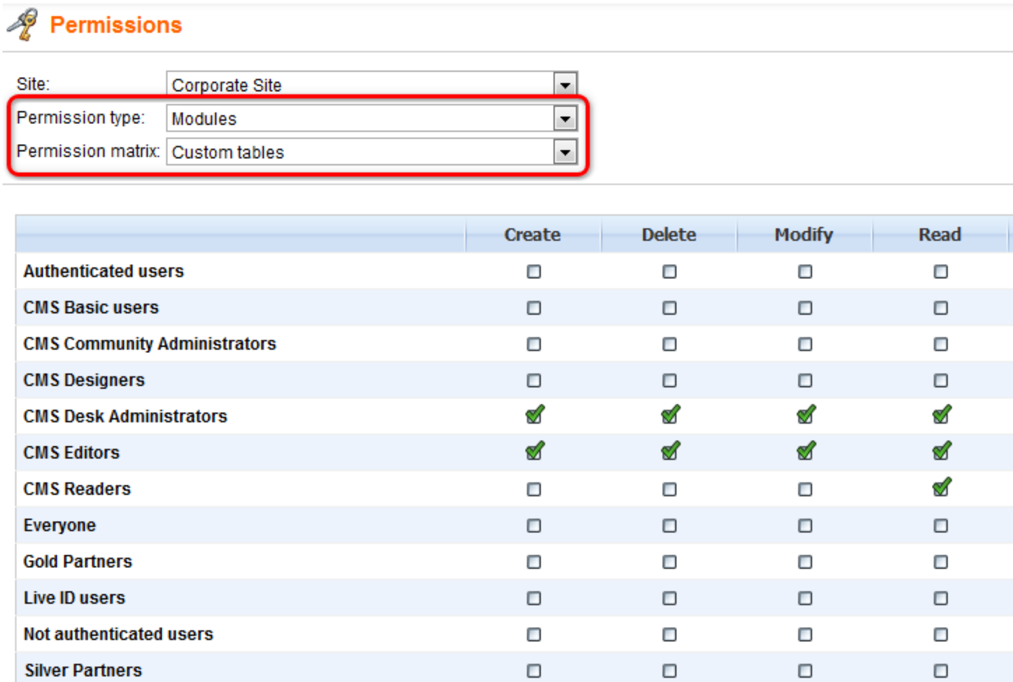
8.14.7 Security

Permissions of the Custom tables module can be configured on two levels - globally for all custom tables belonging to the current site and separately for each particular custom table. The first, global level is configuration via the **Modules -> Custom tables** permission matrix in **Site Manager -> Administration -> Permissions**.

The following permissions can be granted to particular roles on this level:

- **Create** - members of the role are allowed to create data in any custom table

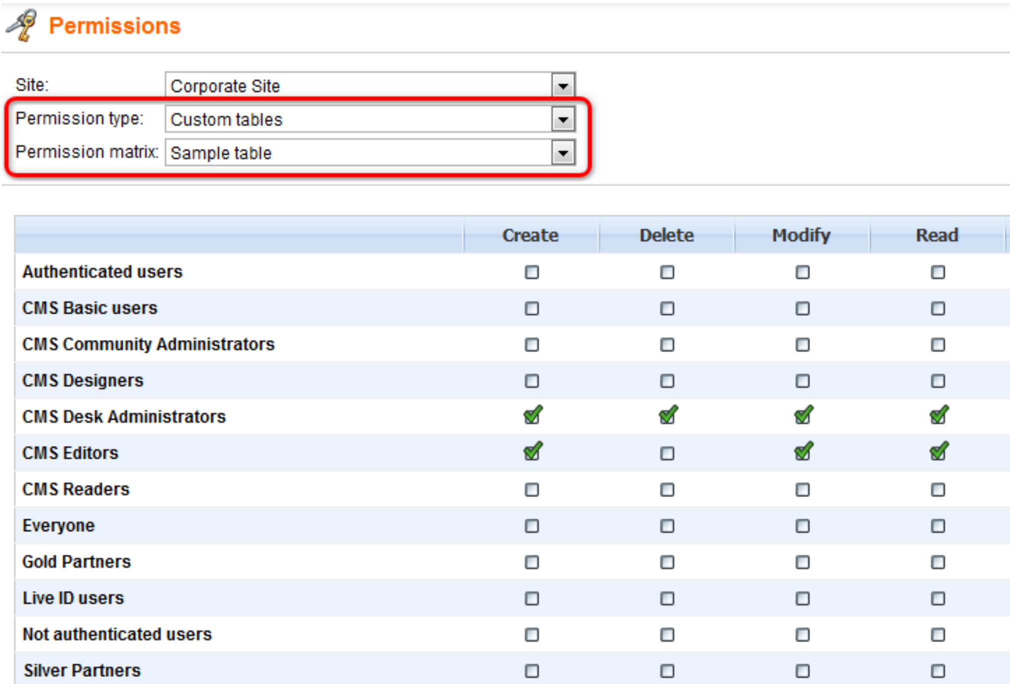
- **Delete** - members of the role are allowed to delete data in any custom table
- **Modify** - members of the role are allowed to modify data in any custom table
- **Read** - members of the role are allowed to read any custom table data



You can also configure these permissions separately for particular custom tables. This can also be done in **Site Manager -> Administration -> Permissions**, but in this case, you have to select **Custom tables** from the **Permission type** drop-down list. The **Permission matrix** drop-down list will then offer you all custom tables assigned to the selected site.

The permissions have the same names and meanings as the global ones, but this time, they are related only to the one selected custom table:

- **Create** - members of the role are allowed to add new records into the selected table
- **Delete** - members of the role are allowed to delete records from the selected table
- **Modify** - members of the role are allowed to modify existing records in the selected table
- **Read** - members of the role are allowed to read data stored in the selected table



	Create	Delete	Modify	Read
Authenticated users	☐	☐	☐	☐
CMS Basic users	☐	☐	☐	☐
CMS Community Administrators	☐	☐	☐	☐
CMS Designers	☐	☐	☐	☐
CMS Desk Administrators	☑	☑	☑	☑
CMS Editors	☑	☐	☑	☑
CMS Readers	☐	☐	☐	☐
Everyone	☐	☐	☐	☐
Gold Partners	☐	☐	☐	☐
Live ID users	☐	☐	☐	☐
Not authenticated users	☐	☐	☐	☐
Silver Partners	☐	☐	☐	☐

8.14.8 Custom tables internals and API

8.14.8.1 Database tables and API classes

The Custom tables module uses the following database tables:

- **CMS_Class** - custom tables are stored as classes in the system database, this database table is used for storing information about classes
- **CMS_ClassSite** - this table is used to assign classes (and therefore also custom tables) to particular sites in the system
- **CMS_Query** - SQL queries for operations with data in the custom table are stored in this table, along with other queries used by the system
- **CMS_Transformation** - transformations for custom tables data are stored in this table, as well as other transformations used by the system
- **CMS_AlternativeForm** - alternative forms for particular custom tables are stored in this table, together with all other alternative forms used by the system
- **<prefix>_<table code name>** - particular custom tables are stored as standard tables in the system database; you can distinguish them by giving them a descriptive prefix, while the *customtable_* prefix is used by default

The Custom tables API is provided by the following **CMS.SiteProvider** namespace classes:

- **CustomTableItem, CustomTableItemProvider** - these classes provide functionality for managing data items stored in custom tables

The following topics show how methods from these classes can be used to manage custom tables data:

- [Adding an item into a custom table]
- [Getting items from a custom table]
- [Editing a custom table item]

- Deleting a custom table item
- A complete example

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.14.8.2 Adding an item into a custom table

The following code sample shows how a new data item can be added to an existing custom table.

[C#]

```
using CMS.CMSHelper;
using CMS.SettingsProvider;
using CMS.SiteProvider;
using CMS.DataEngine;

...

        string customTableClassName = "customtable.SampleTable";

        // Get data class using custom table name
        DataClassInfo customTableClassInfo = DataClassInfoProvider.GetDataClass(cust
        if (customTableClassInfo == null)
        {
            throw new Exception("Given custom table does not exist.");
        }

        // Initialize custom table item provider with current user info and
general connection
        CustomTableItemProvider ctiProvider = new CustomTableItemProvider(
CMSContext.CurrentUser, ConnectionHelper.GetConnection());

        // Create new custom table item for given class of custom table
        CustomTableItem item = new CustomTableItem(customTableClassInfo.ClassName, c

        // Set value of a custom table item field
        item.SetValue("ItemText", "Sample item text");

        // Insert the item
        item.Insert();
```

8.14.8.3 Getting items from a custom table

The following code example shows how a *DataSet* with data items stored in a custom table can be retrieved.

[C#]

```
using System.Data;
using CMS.CMSHelper;
using CMS.SiteProvider;

...

        string customTableClassName = "customtable.SampleTable";
        string where = "ItemID > 5";
        string orderBy = "ItemOrder";
        int topN = 10;
        string selectedColumns = "ItemID, ItemCreatedBy, ItemModifiedBy";

        // Create an instance of the Custom table item provider
        CustomTableItemProvider tableProvider = new CustomTableItemProvider(CMSConte

        // Get a list of custom table items according to a WHERE condition
        DataSet ds = tableProvider.GetItems(customTableClassName, where, orderBy);

        DataSet ds2 = tableProvider.GetItems(customTableClassName, where, orderBy, t
```

8.14.8.4 Editing a custom table item

The code example below demonstrates how an existing item in a custom table can be edited.

[C#]

```
using CMS.CMSHelper;
using CMS.SettingsProvider;
using CMS.SiteProvider;
using CMS.DataEngine;

...

        string customTableClassName = "customtable.SampleTable";

        // Get data class using custom table name
        DataClassInfo customTableClassInfo = DataClassInfoProvider.GetDataClass(cust
        if (customTableClassInfo == null)
        {
            throw new Exception("Given custom table does not exist.");
        }

        // Initialize custom table item provider with current user info and
general connection
        CustomTableItemProvider ctiProvider = new CustomTableItemProvider(
CMSContext.CurrentUser, ConnectionHelper.GetConnection());

        // Provide ID of item you want to edit
        int itemId = 1;
```

```
        // Get custom table item with given ID
        CustomTableItem item = ctiProvider.GetItem(itemId, customTableClassInfo.Clas

        // Set value of the custom table item field
        item.SetValue("ItemText", "Sample item");

        // Update item
        item.Update();
```

8.14.8.5 Deleting a custom table item

The code example on this page shows how an existing item can be deleted from a custom table.

[C#]

```
using CMS.CMSHelper;
using CMS.SettingsProvider;
using CMS.SiteProvider;
using CMS.DataEngine;

...

        string customTableClassName = "customtable.SampleTable";

        // Get data class using custom table name
        DataClassInfo customTableClassInfo = DataClassInfoProvider.GetDataClass(cust

        // Check if data class info exists
        if (customTableClassInfo != null)
        {
            // Initialize custom table item provider with current user info and gene
            CustomTableItemProvider ctiProvider = new CustomTableItemProvider(
CMSContext.CurrentUser, ConnectionHelper.GetConnection());

            // Provide ID of item you want to delete
            int itemId = 1;

            // Get custom table item with given item ID
            CustomTableItem item = ctiProvider.GetItem(itemId, customTableClassInfo.

            // Check if item exists
            if (item != null)
            {
                // Delete item
                item.Delete();
            }
        }
```

8.14.8.6 Getting custom tables

In the following code example, you can see how to get a *DataSet* with all custom tables in the system.

[C#]

```
using CMS.CMSHelper;
using CMS.SettingsProvider;
using CMS.SiteProvider;
using CMS.DataEngine;

...

        // Retrieve all custom tables from the database
        DataSet customTables = DataClassInfoProvider.GetCustomTableClasses();
```

This second code example demonstrates how only custom tables matching the specified criteria can be retrieved and saved in a *DataSet*.

[C#]

```
using CMS.CMSHelper;
using CMS.SettingsProvider;
using CMS.SiteProvider;
using CMS.DataEngine;

...

        string where = "ClassID > 5";
        string orderBy = "ClassDisplayName";
        int topN = 10;
        string columns = "ClassID, ClassDisplayName, ClassLastModified";

        // Get custom tables according to the given criteria
        DataSet customTables2 = DataClassInfoProvider.GetCustomTableClasses(
CMSContext.CurrentSiteID, where, orderBy, topN, columns);
```

8.14.8.7 A complete example

The following code example consists of a sequence of custom table data item management operations. A data item is first created and inserted into a custom table. After that, one of its columns is modified with different data. Next, the order of the item is modified. Finally, the item is deleted from the table and in the very end, all items in the custom table are deleted as well.

[C#]

```
using System;
using System.Data;
using CMS.SiteProvider;
using CMS.CMSHelper;
```

```
using CMS.GlobalHelper;

...

        string tableName = "customtable.sampletable";

        // Create an instance of the Custom table item provider
        CustomTableItemProvider tableProvider = new CustomTableItemProvider(CMSConte

        // Create a custom table item object
        CustomTableItem cti = new CustomTableItem(tableName, tableProvider);
        // Set the ItemText field value
        cti.SetValue("ItemText", "Sample text");
        // Insert the item object into the database
        cti.Insert();

        // Get an item from a specified custom table
        cti = tableProvider.GetItem(1, tableName);

        if (cti != null)
        {
            // Get names of the custom table columns
            string[] columns = cti.ColumnNames;

            // If the table contains the ItemText column, get its value
            if (cti.ContainsColumn("ItemText"))
            {
                string text = ValidationHelper.GetString(cti.GetValue("ItemText"), "

                // Modify the column value
                text += " modified";
                cti.SetValue("ItemText", text);

                // Update the database record
                cti.Update();
            }

            // Check if the custom table items can be ordered
            if (cti.OrderEnabled)
            {
                // Move the item down
                tableProvider.MoveItemDown(cti.ItemID, tableName);

                // Move the item up
                tableProvider.MoveItemUp(cti.ItemID, tableName);
            }

            // Delete the custom table item from the database
            cti.Delete();

        }

        // Delete all items from a specified custom table
        tableProvider.DeleteItems(tableName);
```

8.15 Document library

8.15.1 Overview

The Document library module enables live site users to upload and manage files stored in a pre-defined location in the content tree. The module has no dedicated user interface. Its whole functionality is provided by two web parts: **Document library** and **Group document library**. Both web parts and their specific properties are described in the Document library web parts topic. The module also comes with the Document library widget, which is based on the **Document library** web part.

Document library files are stored in the content tree as standard *CMS.File* documents. This means that even though the module has no dedicated user interface, actions made on the live site are reflected in the content tree. Like this, the uploaded documents can also be managed by website editors or administrators in **CMS Desk -> Content**. Only flat file structure of document libraries is currently supported, i.e. you can not store files in sub-folders but only under the main document. More information about *CMS.File* documents, as well as other ways how files can be stored in Kentico CMS, can be found in Content management -> File management.

A live example of use of this module can be found on the **Intranet Portal** sample website, specifically on its */Documents* page. The process of creating a similarly structured document library can be found in the Creating a document library topic. The Managing files in document libraries topic explains how the module can be used from the live site user's point of view. Finally, the Security topic explains how standard document-level permissions are applied to documents in the library and how they can be configured.

8.15.2 Document library web parts

The whole functionality of the Document library module is ensured by the following two web parts:

- **Document library -> Document library** - designed for general use anywhere on your website; it displays all documents from a specified location except those whose *Owner* is a community group
- **Community -> Group document library** - designed for use within a context of a community group; displays documents whose *Owner* is the specified community group along with other documents stored under the specified path

The web parts have the following specific properties:

- **Library path** - path to the document under which the library documents are stored, e.g. */My-document-library/My-images*; only *CMS.File* documents from the first level under the entered path are displayed; using the **Set permissions** button, you can directly configure the document's

permissions, as described in Security
- **Page size** - number of documents listed by the web part on a single page
- **Document form** - Alternative form that will be used for document properties editing (after choosing **Properties** from the context menu); if blank, the *CMS.File.DocumentLibrary* form will be used; if this form is not available, the *CMS.File* document type's default form will be used
- **Group name** - available only with the *Group document library* web part; documents owned by the specified group stored under the *Library path* will be displayed along with other documents under the same path
- **Combine with default culture** - if *No*, only documents from the current culture that are stored under the *Library path* will be displayed; if *Yes*, documents from the default culture will be displayed along with documents from the current culture, while default culture documents will be marked with the culture's flag (as can be seen in the screenshot below) and the **Translate** option will be available in their context menu
- **Check permissions** - if enabled, only documents for which the current user has the **Read** permission are displayed by the web part; if disabled, all documents are displayed

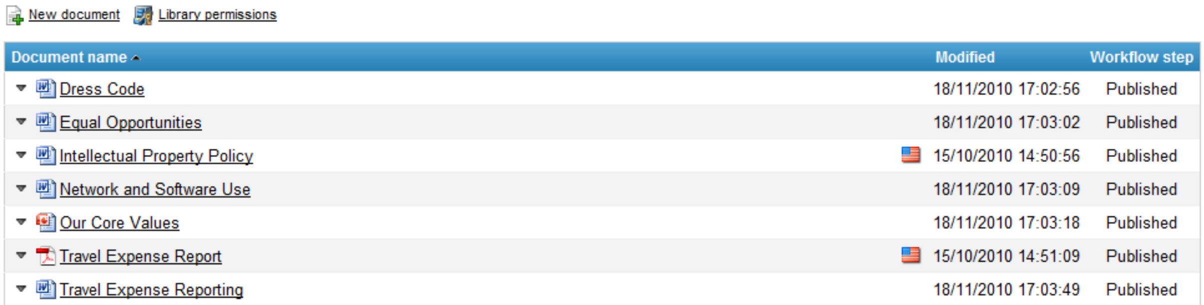
8.15.3 Document library widget

The module also comes with the **Document library** widget. The widget is stored under the **Document library** widget category and it is an editor widget, i.e. it can only be added to editor widget zones of page templates by website editors in CMS Desk.

Compared to the Document library web parts, the **Document library** widget only has a limited set of properties to configure:

- **Library path** - path to the document under which the library documents are stored, e.g. */My-document-library/My-images*; only *CMS.File* documents from the first level under the entered path are displayed; using the **Set permissions** button, you can directly configure the document's permissions, as described in Security
- **Page size** - number of documents listed by the web part on a single page
- **Combine with default culture** - if *No*, only documents from the current culture that are stored under the *Library path* will be displayed; if *Yes*, documents from the default culture will be displayed along with documents from the current culture, while default culture documents will be marked with the culture's flag (as can be seen in the screenshot below) and the **Translate** option will be available in their context menu

Besides the fact that only a limited set of properties is available with the widget, its functionality is identical to the **Document library** web part.



Document name		Modified	Workflow step
Dress Code		18/11/2010 17:02:56	Published
Equal Opportunities		18/11/2010 17:03:02	Published
Intellectual Property Policy		15/10/2010 14:50:56	Published
Network and Software Use		18/11/2010 17:03:09	Published
Our Core Values		18/11/2010 17:03:18	Published
Travel Expense Report		15/10/2010 14:51:09	Published
Travel Expense Reporting		18/11/2010 17:03:49	Published

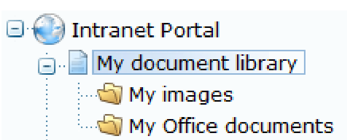
8.15.4 Creating a document library

This topic explains how a document library can be created on your website. We will create a sample page with two **Document library** web parts, each of which will display files from a separate folder. Like this, we will actually have two document libraries on a single page, each for storing different types of files. The example is based on the **Intranet Portal** sample website. However, the procedure is analogical when performed on any other website.

The first step when creating a document library is to choose a location in the content tree where the files will be stored. A single document library web part can display documents from only a single location. As we want two separate libraries on a single page, we will create a page in the content tree where two **Document library** web parts will be placed. Under the page, we will create two separate folders where the actual files will be stored.

1. Go to **CMS Desk -> Content**, select the root of the content tree and click **New**. Choose the **Page (menu item)** document type. In the following dialog, enter *My document library* into the **Page name** field and choose to **Create a blank page** using the **Simple** layout. Click **Save**.

With the new document selected in the content tree, click **New** again. This time, choose the **Folder** document type and enter *My images* as its **Document name**. Repeat the same once again to create another folder, while this one should be called *My Office documents*. The final result should look as in the screenshot below.



2. Now that you have the page and folders created, we may proceed to adding the **Document library** web parts. Select the **My document library** document, switch to the **Design** mode and click the **Add web part** () icon of the only web part zone on the page. In the **Select web part** pop-up dialog, choose the **Document library** web part, which is stored in the **Document library** web part category. Configure its properties as follows:

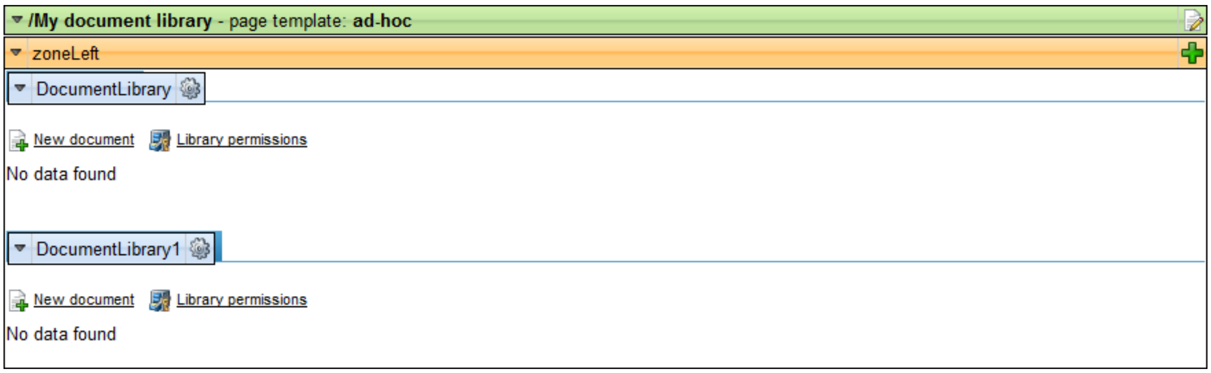
- **Library path**: /My-document-library/My-images
- **Page size**: 15
- **Document form**: leave the field blank
- **Web part container**: Intranet simple container (with header line)
- **Container title**: My images
- **Content after**:

Click **OK**.

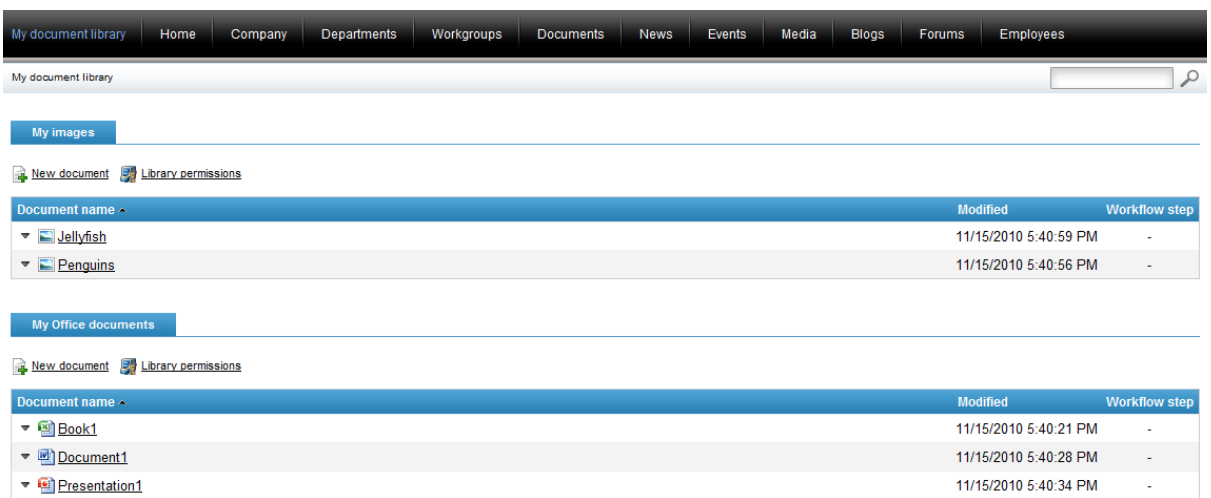
3. Repeat step 2 and add another **Document library** web part to the page. This time, use the following values of its properties:

- **Library path**: /My-document-library/My-Office-documents
- **Page size**: 15
- **Document form**: leave the field blank
- **Web part container**: Intranet simple container (with header line)
- **Container title**: My Office documents

Click **OK**. The web part zone should now look as in the following screenshot if viewed in the **Design** mode.



4. The document libraries are now configured and ready to be used by live site users. If viewed on the live site after uploading some files into the libraries, the page should look as in the screenshot below.



Please proceed to the Managing files in document libraries topic to learn more about how live site users can upload and manage document library files.

8.15.5 Managing files in document libraries

This topic explains particular actions that can be performed by live site users in a document library.

Permissions for particular actions

All of the actions listed below **are not always available**. They are displayed based on document permissions granted to the current user or their roles. Please consult the Security topic for more details on which permissions are required for each of the listed actions.

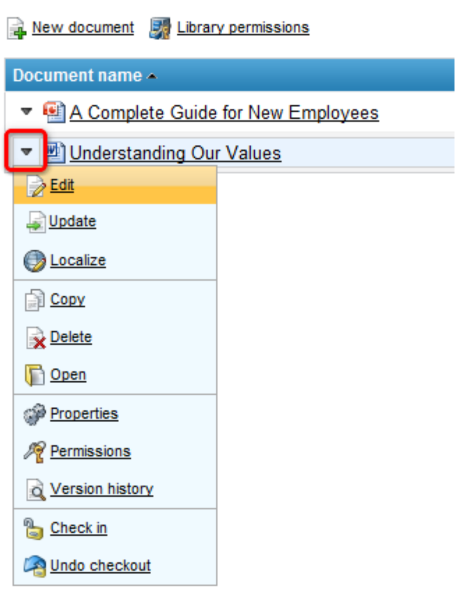
When a new document library is created, there are no files in it, so live site users can only see the web part with the following two links at the top.

- New document
- Library permissions

Once some files are uploaded in the document library, live site users can perform the following actions with the documents:

- Edit
- Update
- Localize
- Copy
- Delete
- Open
- Properties
- Permissions
- Version history
- Workflow actions

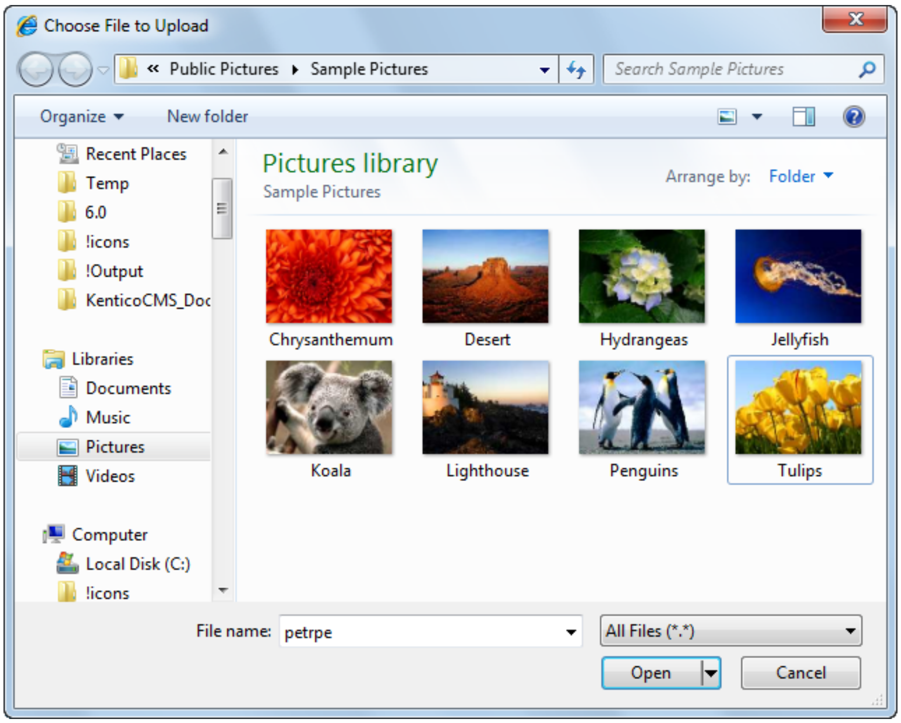
These actions can be executed from a context menu, which is accessible by right-clicking anywhere in a document's row, or by left-clicking the down-arrow to the left of a document's name (as highlighted in the screenshot below).



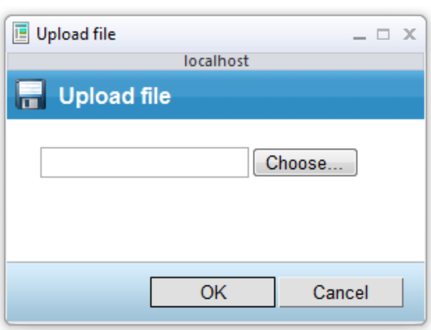
The following text explains particular actions that can be performed in a document library:

New document

Clicking the **New document** link in **Internet Explorer** or **Mozilla Firefox** directly opens the browser's file selection dialog. By selecting a file from a local disk and clicking **Open**, the file will be uploaded into the document library.

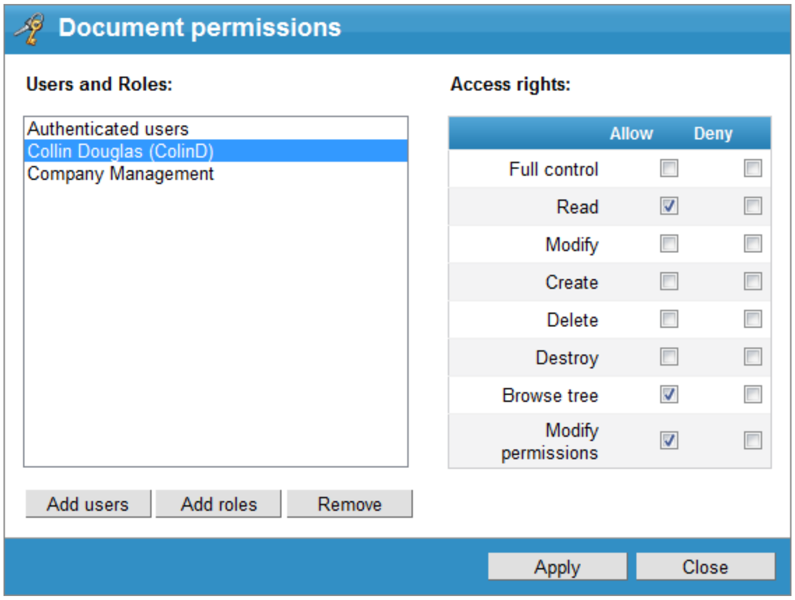


When uploading new files in **Opera** or **Apple Safari**, there is one extra step in the procedure: the **Upload file** dialog gets displayed first after clicking the **New document** link. Clicking the **Choose** button in the dialog opens the browser's file selection dialog, which can be subsequently used as described above. The screenshot below shows the **Upload file** dialog viewed in **Opera.** In **Apple Safari**, the dialog may look slightly different.



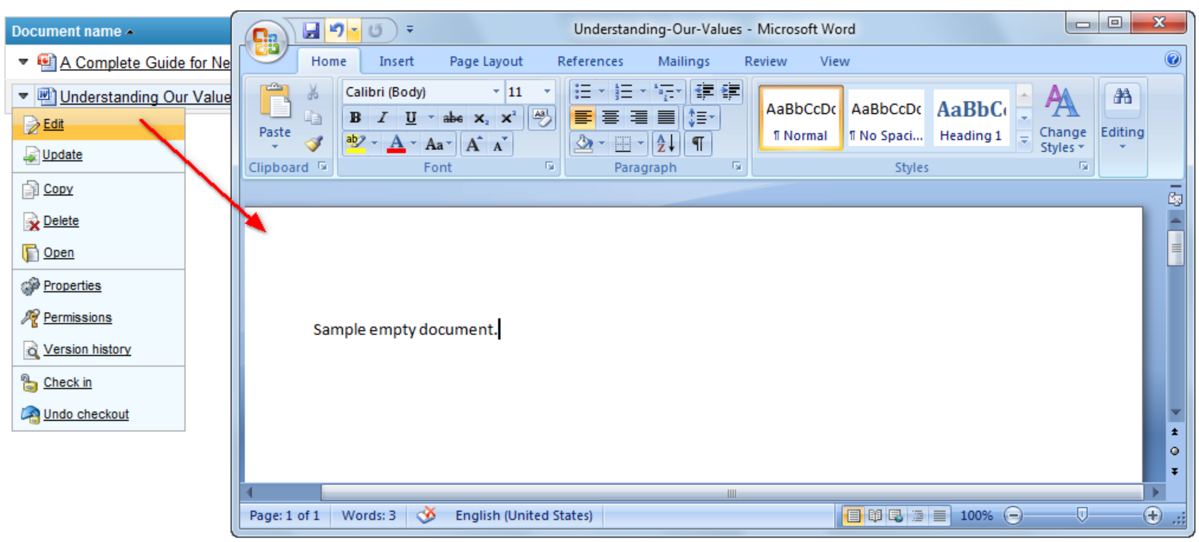
Library permissions

Clicking the **Library permissions** link opens a dialog for document library permissions configuration. Please refer to the **Configuring document-level permissions on the live site** section of the Security topic for more details.



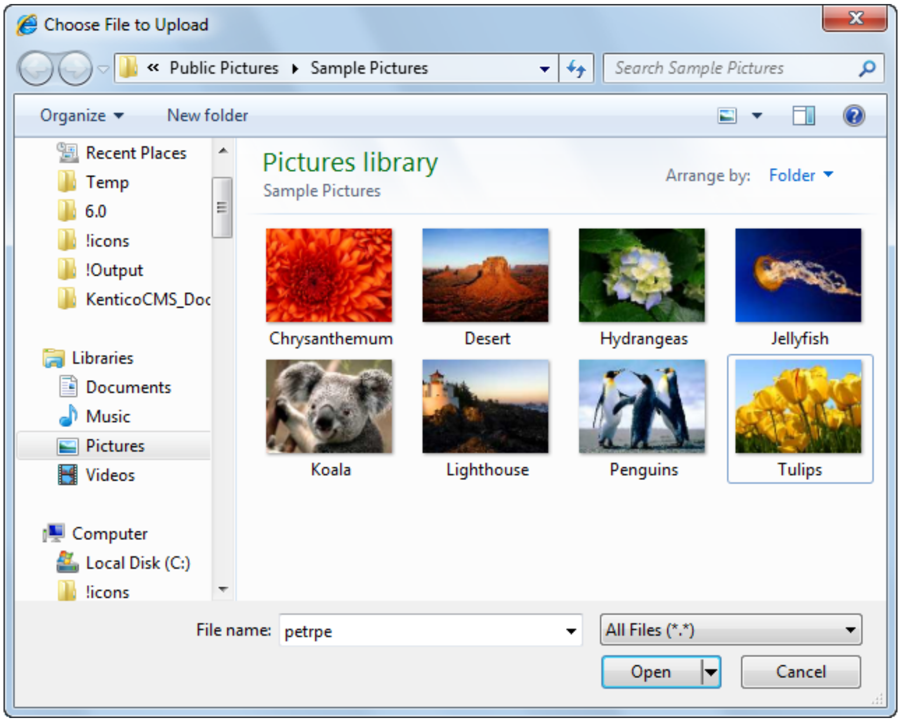
Edit

The **Edit** action is only available if WebDAV integration is enabled. It opens the file in a client application for direct editing (for example, *.docx* documents get opened in *MS Word*, as you can see in the screenshot below). WebDAV editing is only possible if there is an application installed on the client machine that supports WebDAV editing of the particular file type.



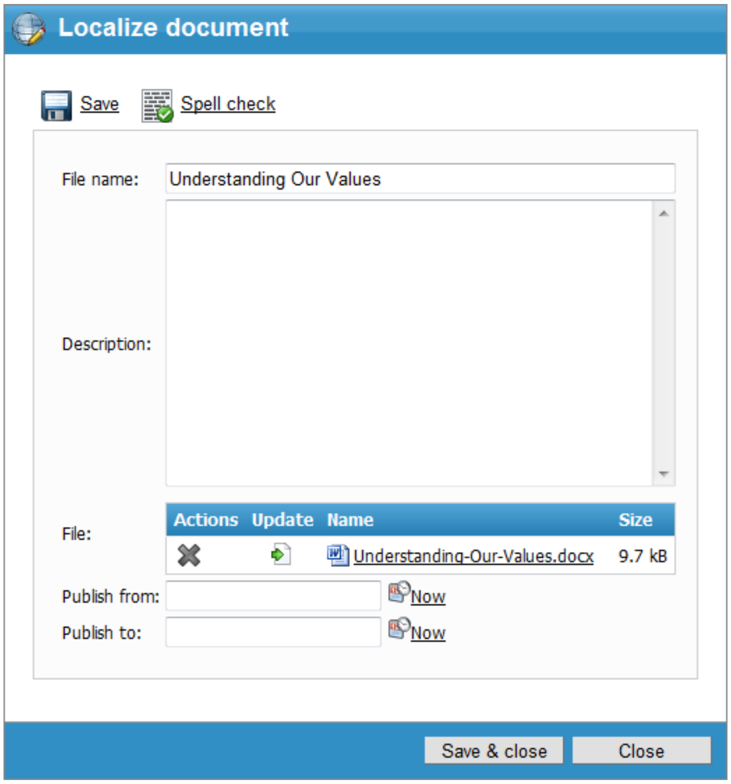
Update

Selecting the **Update** action opens the browser's file selection dialog. Selecting a file from a local disk and clicking **Open** in the dialog replaces the original file with the selected one. In **Apple Safari** and **Opera**, there is one extra dialog displayed before the browser's file selection dialog. Please refer to the New document paragraph above for more details.



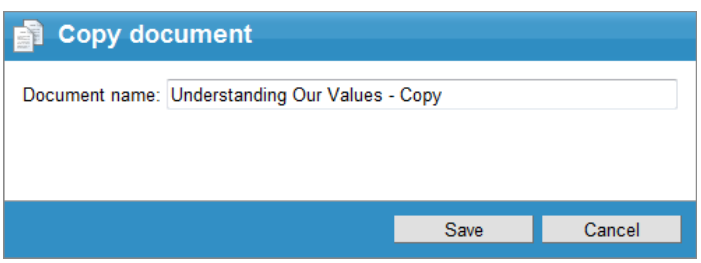
Localize

The **Localize** option is only displayed with documents from the default culture when the **Combine with default culture** property of the web part is enabled. Clicking the action opens the dialog depicted in the screenshot below, which serves for creation of a new language version of the selected document in the current culture.



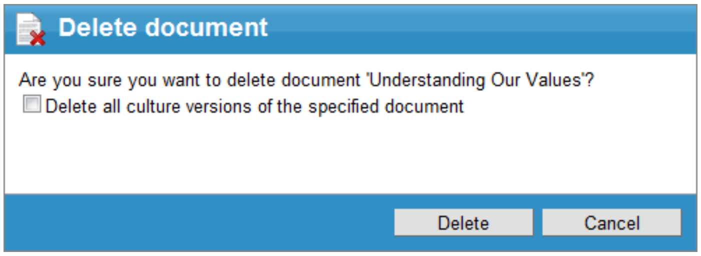
Copy

The **Copy** action creates a copy of the file in the same document library. As only flat structure of document libraries is currently possible, a different file name needs to be entered for the copy, while the " - Copy" suffix is offered to be added to the file name automatically. When copying a linked document (marked with the flag), the copy will be a standard document, not a linked one.



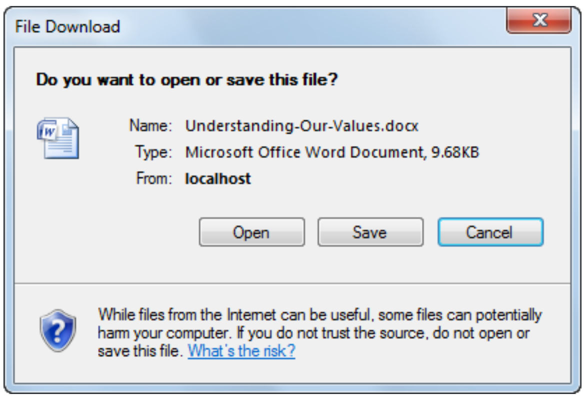
Delete

Clicking the **Delete** action opens the dialog depicted below. Using the dialog, you can remove the document from the document library. On multilingual websites where more that one language version of the document exists, the **Delete all culture versions of the specified document** check-box is displayed in the dialog. If you delete the document with the check-box enabled, all cultural versions of the document will be deleted.



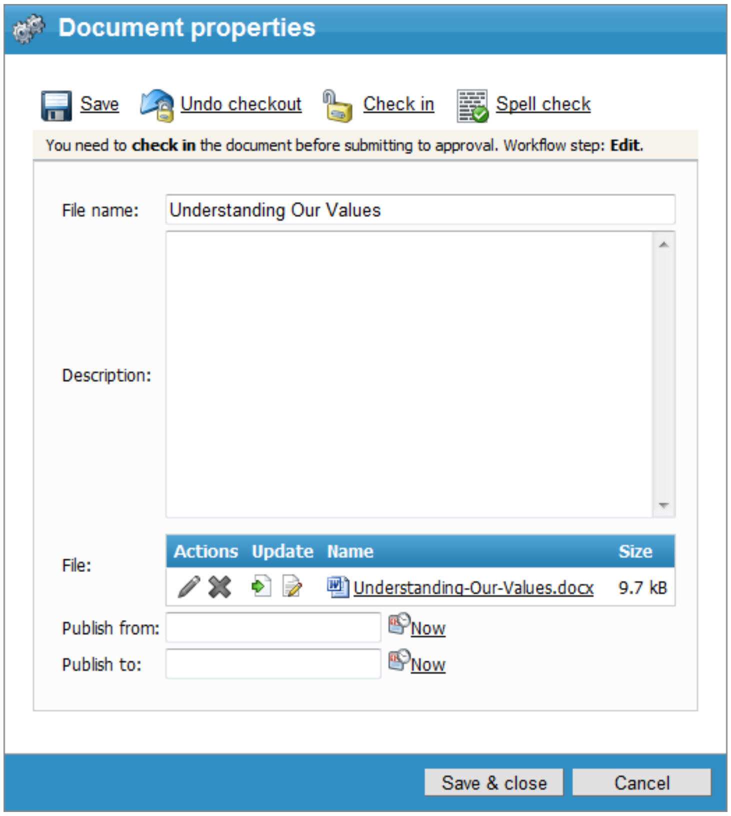
Open

The **Open** action works similarly as a standard file download link. The web browser's standard file download dialog is displayed and you can either open the document or save it to a local disk. The same behavior can be expected when you click the document's name in the library listing.



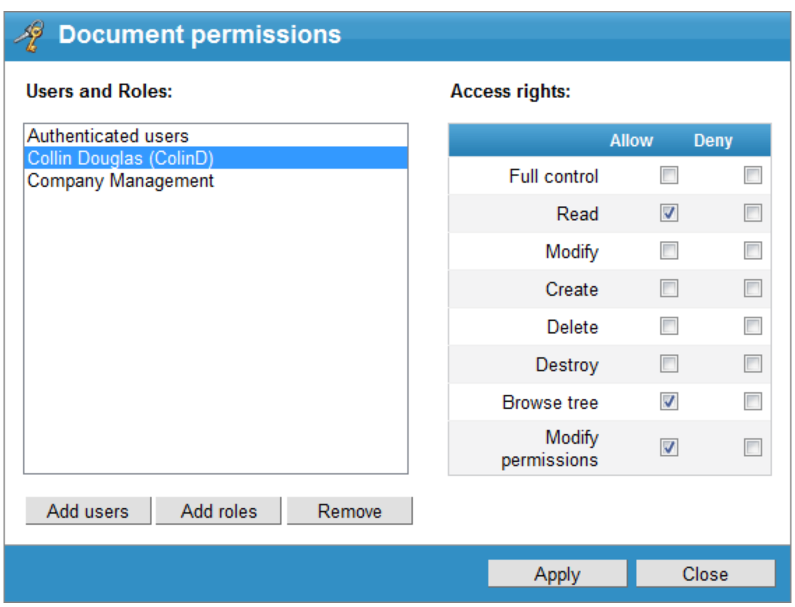
Properties

The **Properties** action opens a dialog where properties of the *CMS.File* can be configured the same way as in **CMS Desk -> Content -> Edit -> Form**. The dialog uses the Alternative form form specified in the **Document form** property of the web part. If no form is specified, the *CMS.File.DocumentLibrary* form is used. If even this form is not available, *CMS.File* document type's default form is used.



Permissions

The **Permissions** action opens a dialog for configuration of permissions for the particular document, similarly as in **CMS Desk -> Content -> Edit -> Properties -> Security**. Please refer to the **Configuring document-level permissions on the live site** section of the Security topic for more details.



Version history

Clicking the **Version history** action displays an overview of the document's versions, similarly as in **CMS Desk -> Content -> Edit -> Properties -> Versions**. The following actions are available with each of the listed versions:

- **View -** displays an overview of the document's content with the possibility of side-by-side comparison of versions
- **Rollback -** rolls back any changes made since the particular version of the document
- **Delete** - deletes the old version

Clicking the **Destroy history** button deletes all versions of the document except the current one.

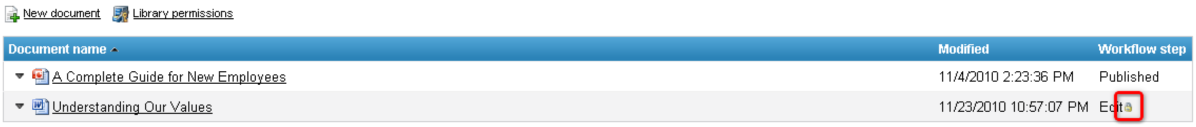
Workflow actions

If there is a workflow configured for documents in a document library, standard workflow actions are displayed at the bottom of the context menu, depending on workflow configuration and the current workflow step of the document:

- **Submit to approval** - submits the document for approval to the sub-sequent workflow step
- **Approve**[1] - approves the document and switches it to the sub-sequent workflow step
- **Reject**[1] - rejects the document and switches it back to the previous workflow step
- **Archive** - switches the document to the Archived step
- **Check out**[2] **-** checks the document out for editing; while checked out, the document can not be edited by other users; checked-out documents are marked with the icon in the document library listing (as can be seen in the screenshot below)
- **Check in**[2] - checks the modified document in after editing, so that other users can edit it again
- **Undo checkout**[2] - takes back the check-out, while changes made to the document are not saved

[1] for these actions to be available, the user must also be in one of the roles that are allowed to approve/reject the document in the current workflow step or have the **Manage workflow** permissions for all content
[2] these actions are only available if the workflow applied to the document is configured to use check-in/check-out

Document libraries are typically used with Versioning without workflow. In this case, document in the libraries are in the **Published** workflow step all the time and a new version of the document is created with each modification. Please refer to Development -> Workflow and versioning for more information on workflows in Kentico CMS.



8.15.6 Security

The Document library module leverages standard document permissions. The following table explains which permissions are required to perform particular actions in the document library. These permissions can be granted to roles globally for all content or for the *CMS.File* document type, or to particular users or roles on document-level of the library's parent document or each particular document in the library.

Action	Read	Modify	Create	Delete	Destroy	Manage workflow	Modify permissions
New document	•	•	•				
Library permissions	•						•
Edit	•	•					
Update	•	•					
Localize	•	•	•				
Copy	•	•	•				
Delete	•			•			
Open	•						
Properties	•	•					
Permissions	•						•
Version history	•	•			1		
Submit to approval	•	•					
Approve[2]	•	•				2	
Reject[2]	•	•				2	
Archive	•	•				•	
Check out[3]	•	•					
Check in[3]	•	•					
Undo checkout[3]	•	•					

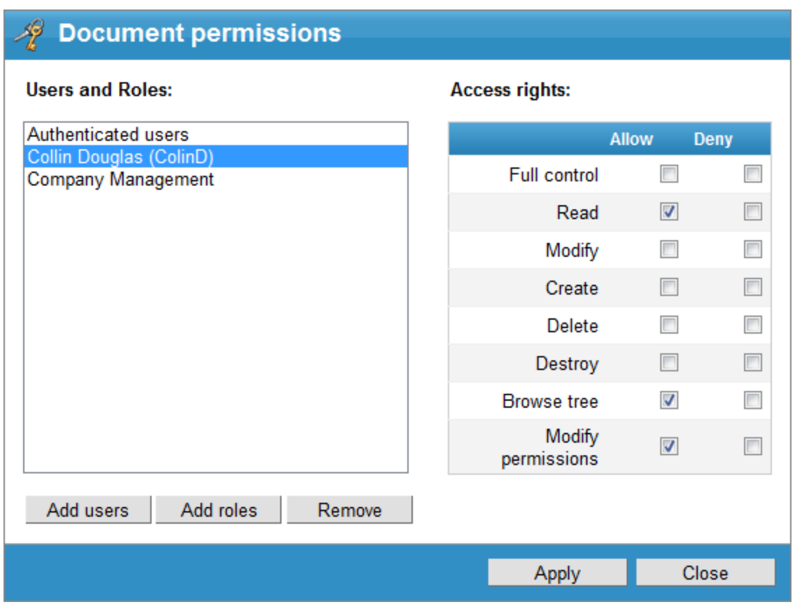
[1] The **Destroy** permission is required for the user to be able to delete particular versions or the whole version history.
[2] For these actions to be available, the user must also be in one of the roles that are allowed to approve/reject the document in the current workflow step or have the **Manage workflow** permissions for all content.
[3] These actions are only available if the workflow applied to the document is configured to use check-in/check-out.

Configuring document-level permissions on the live site

Document-level permissions can be configured directly on the live site. They can be configured either globally for the document library's parent document, which results in the permissions being inherited by the child documents in the library, or separately for each particular document in the library. Permissions can be granted to users or roles. Permissions for group document libraries can also be granted to group members and group roles.

The **Library permissions** action opens a dialog for configuration of the library's parent document permissions, i.e. the permissions that can be inherited by its child documents (the actual documents stored in the library). This dialog is identical to the **Permissions** section of the **CMS Desk -> Content -> Edit -> Properties -> Security** dialog for the document. Permissions configured via the web part on the live site will be reflected in this dialog.

By choosing the **Permissions** action from the context menu of a document in the library, the same dialog gets displayed, while this time, permissions are configured just for the particular document. Here again, the permissions configured on the live site will be reflected in the **CMS Desk -> Content -> Edit -> Properties -> Security** dialog for the document.



Permissions and workflow

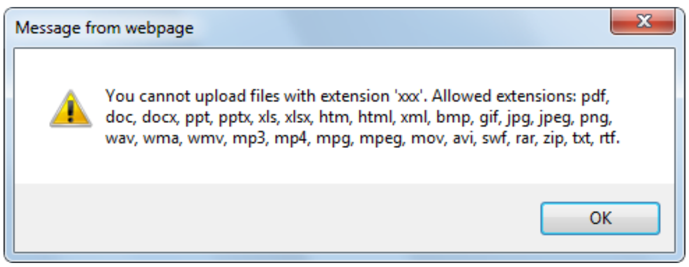
Document libraries reflect workflows applied to documents stored in them. Unless the current user has the **Modify** permission for a document, the currently published version of the document is always displayed to the user. If the document is currently archived or not published, the document is not displayed to the user at all. If the current user does have the **Modify** permission, the current version of the document (in the current workflow step) is displayed to them.

Please refer to Development -> Workflow and versioning for more information on workflows in Kentico CMS.

Allowed file extensions

When uploading a new document into the document library using the **New document** link or updating a document using the **Update** action (both in the context menu and in the **Properties** dialog), only files with extensions defined in **Site Manager -> Settings -> Files -> Upload extensions** or in the **Allowed extensions** property of the *FileAttachment* field of the *CMS.File* document type can be uploaded.

Each attempt to upload a file with an extension that is not allowed will result in the error message as displayed in the screenshot below.



8.16 E-commerce

8.16.1 Overview

For more information about the **E-commerce** module please refer to Kentico CMS E-commerce Guide.

8.17 E-mail queue

8.17.1 Overview

The E-mail queue module was designed to enable the sending of large amounts of e-mails, e.g. when sending newsletter issues to subscribers, without the risk of losing any of the e-mails due to errors. If an e-mail is not delivered successfully, it remains in the queue so that it can be resent later. The e-mails in the queue are sent **automatically**, no manual sending is necessary as this is handled by the **Send queued e-mails** scheduled task.

E-mails in the queue can be monitored or manually managed through the administration interface. Please see the Administrating the e-mail queue topic for more information.

To utilize the e-mail queue, authorized users have the option of Sending mass e-mails to large amounts of users. Certain other modules, such as Newsletters, also make use of the e-mail queue. All other e-mails sent by the system can be configured to either use the e-mail queue or be sent directly to the SMTP server, as is described in the Settings topic.

8.17.2 Administrating the e-mail queue

The e-mail queue administration interface is located in **Site Manager -> Administration -> E-mail queue**. Using the **Site** drop-down list, you can choose which site you want to display the e-mail queue for. If *(global)* is selected, e-mails sent from the administration interface will be displayed. If *(all)* is selected, all e-mails in all queues will be displayed. The administration interface is divided into three tabs:

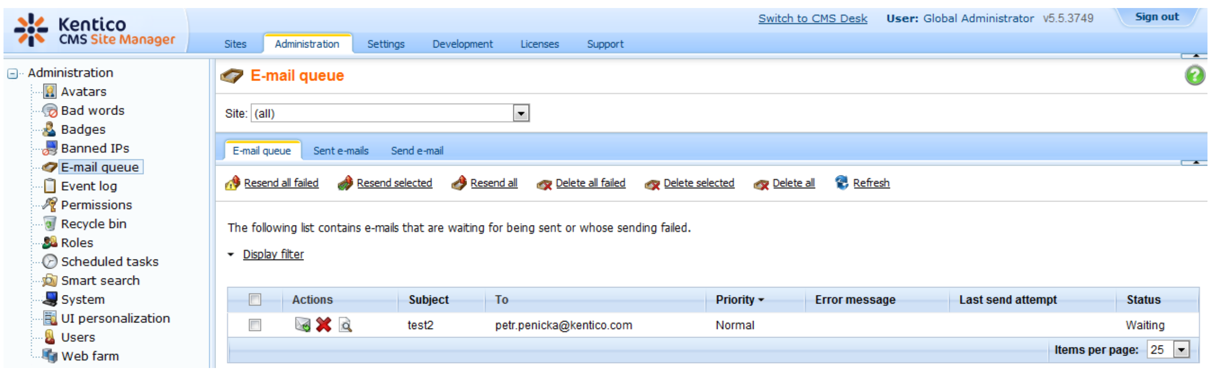
E-mail queue tab

This tab displays the actual e-mail queue. E-mails that are waiting to be sent or that haven't been sent successfully (displayed with an Error message) are displayed here. You can **Resend** ( ) the mail, **Delete** ( ) it or **View** ( ) its details. There is also a number of links above the list:

- **Resend all failed** - resends all e-mails in the queue that were unsuccessfully sent; new e-mails that have not yet been sent will not be resent
- **Resend selected** - resends all e-mails selected by the check-boxes in the list
- **Resend all** - resends all e-mails in the list

- **Delete all failed** - deletes all e-mails in the queue that were unsuccessfully sent; new e-mails that have not yet been sent will not be deleted
- **Delete selected** - deletes all e-mails selected by the check-boxes in the list
- **Delete all** - deletes all e-mails in the list

- **Refresh** - refreshes the content of the e-mail queue

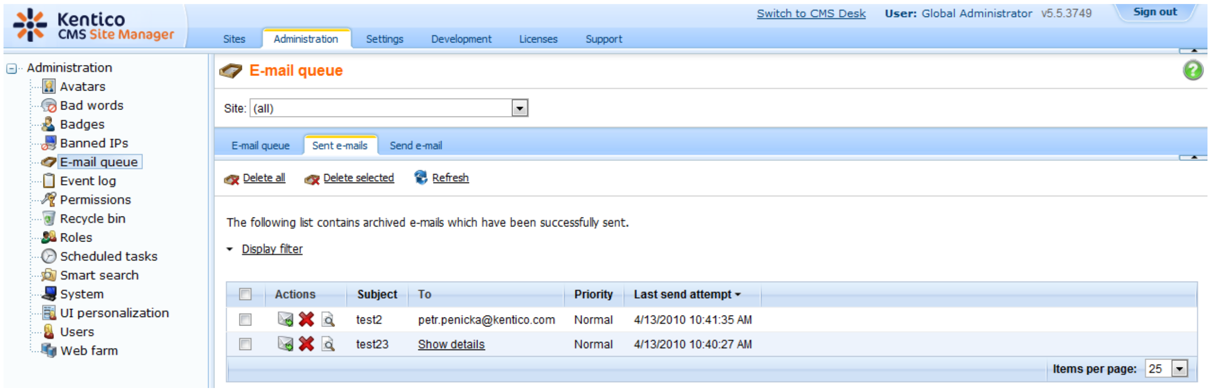


Sent e-mails tab

This tab displays a list of e-mails that have been successfully sent via the e-mail queue. You can set how long the e-mails will stay in this list through the **Site Manager -> Settings -> E-mails -> Archive e-mails (days)** property.

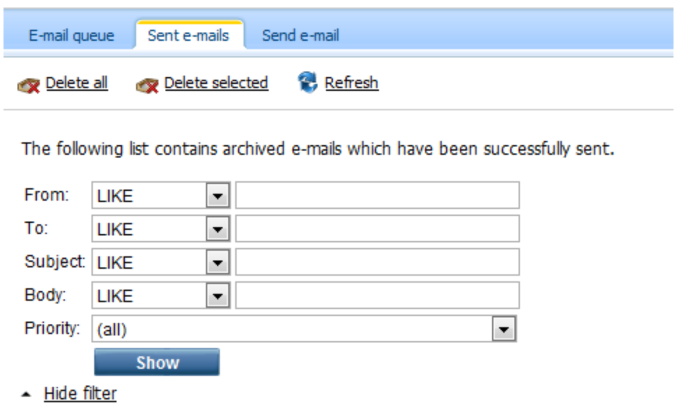
You can **Resend** ( ) the mail, **Delete** ( ) it or **View** ( ) its details. There are also three links at the top of the page:

- **Delete all** - deletes all e-mails in the list
- **Delete selected** - deletes e-mails selected by the check-boxes in the list
- **Refresh** - refreshes the content of the list of sent e-mails



E-mail filter

When there is a large number of e-mails in the e-mail queue, you can easily limit which e-mails will be displayed using the filter. It can be displayed by clicking the **Display filter** link above the e-mail lists on both tabs.



Send e-mail tab

From this tab, you can easily send a single e-mail to a specified recipient (not only the site users). For sending e-mails to multiple users, we recommend using mass e-mails instead of this tab.

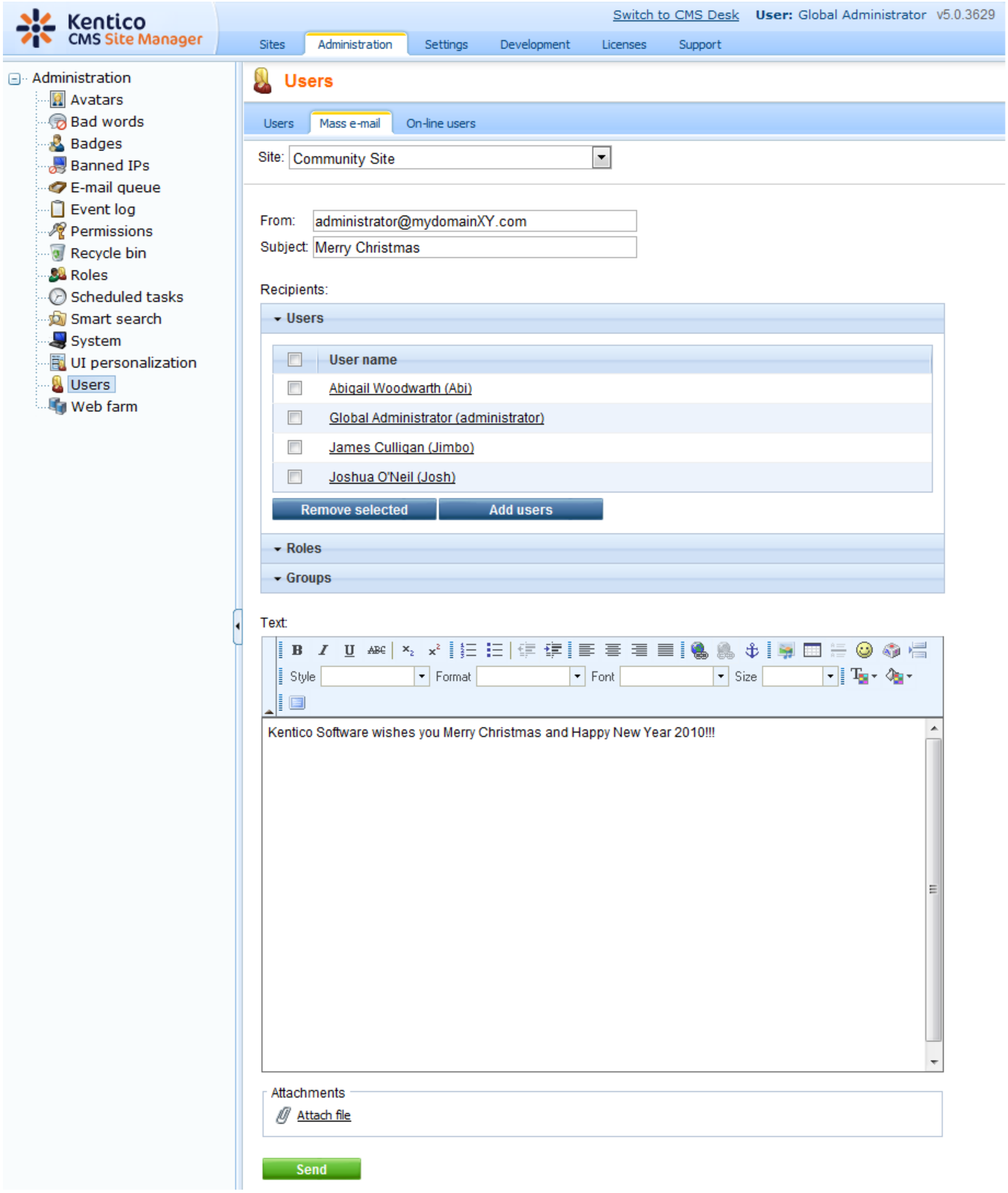
8.17.3 Sending mass e-mails

As the e-mail queue allows the sending of massive amounts of e-mails, we've implemented the Mass e-mail feature. This feature enables you to send the same e-mail to a large amount of users.

You can do this at **Site Manager -> Administration -> Users -> Mass e-mail**.

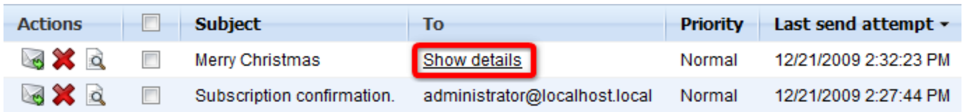
Using the **Site** drop-down list, you can select which site the recipients are related to. Based on this choice, users can be selected either directly, or according to the roles or groups they belong to by using the **Add users** buttons in the **Recipients** section. If you choose **(all)** from the site drop-down list, only the users section will be displayed as roles and groups are always site-related.

You can also add an attachment to the mail using the **Attach file** link.

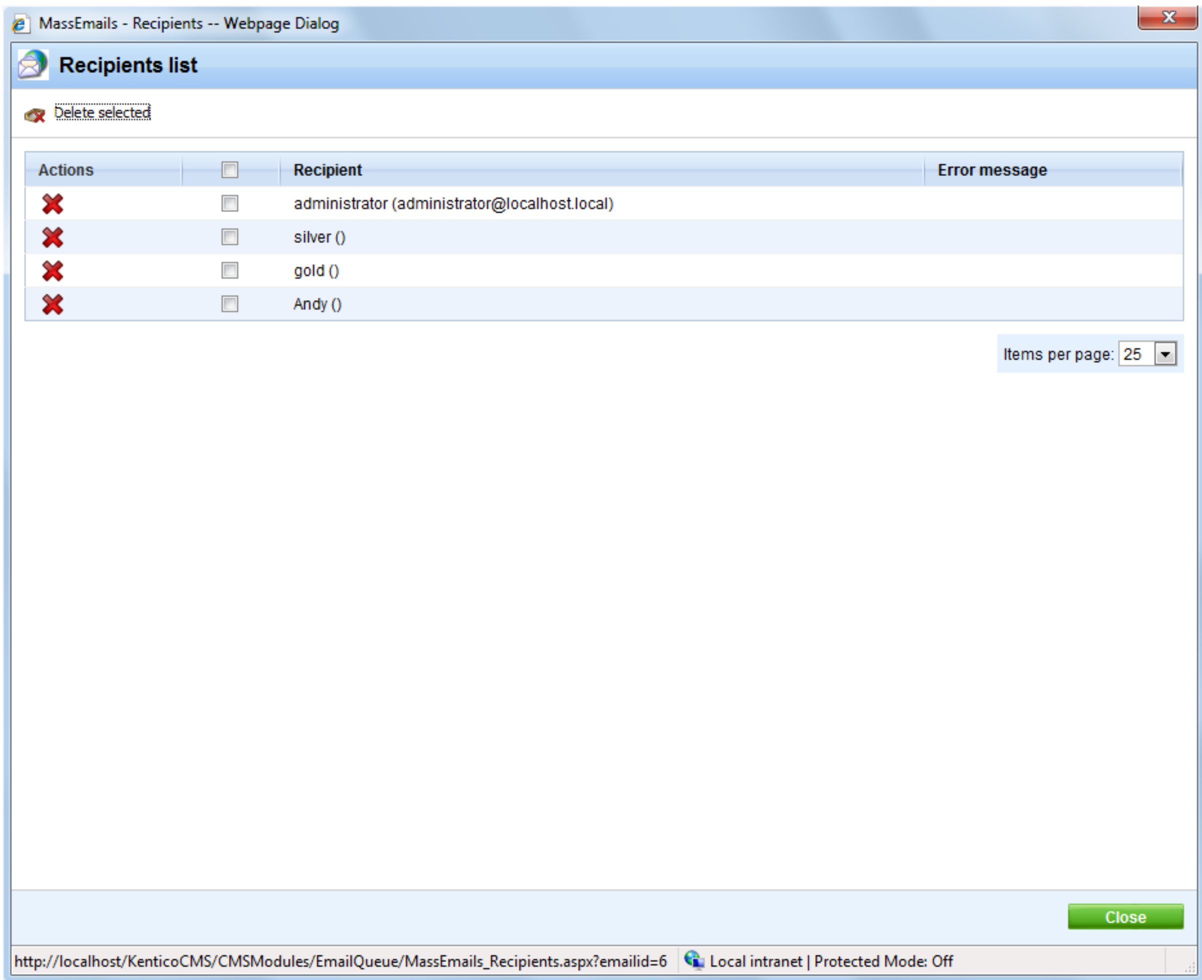


This tab is also available at **CMS Desk -> Administration -> Users -> Mass e-mail**, but only users of the current site can be selected.

In the e-mail queue, you will see the **Show details** link instead of a recipient's e-mail address for mass e-mails, as you can see in the screenshot below.



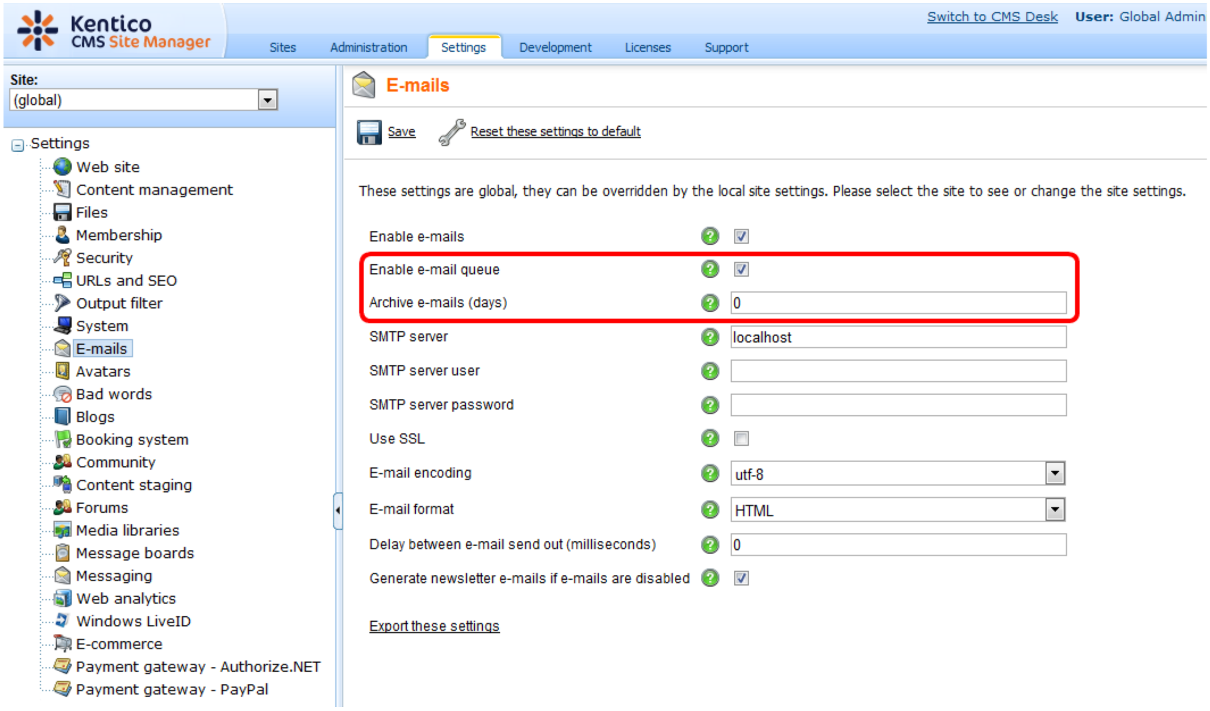
After clicking the link, a window will appear as in the screenshot below. You can delete recipients from the list, so that the mass e-mail will not be delivered to them. This can be done using the **Delete** ( ) icons or by selecting more recipients using the checkboxes and clicking the **Delete selected** link.



8.17.4 Settings

E-mail settings are located at **Site Manager -> Settings -> E-mails**. Among other e-mail related settings, the following two can be used to enable the e-mail queue and configure some of its functionality:

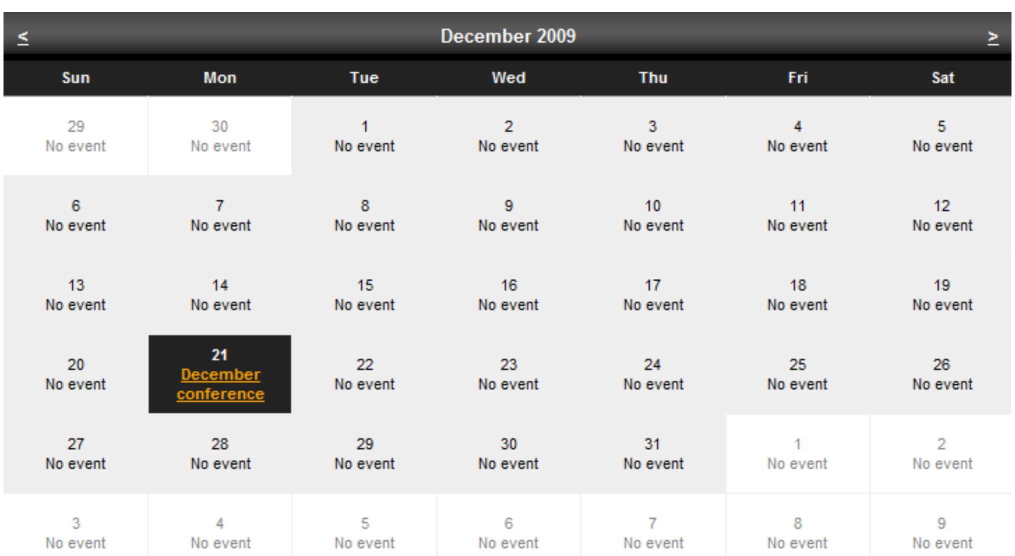
- **Enable e-mail queue** - if checked, the e-mail queue will be used when sending e-mails. There are certain exceptions for priority e-mails, like those containing forgotten passwords for example, that skip the queue and are sent directly
- **Archive e-mails (days)** - the number of days for which e-mails sent via the e-mail queue will remain stored on the **Sent e-mails** tab; if set to 0, e-mails will not be archived



8.18 Event calendar

8.18.1 Overview

The Event calendar web part is used for displaying documents ordered by date in a calendar. Its appearance is similar to the ordinary Calendar web part. You can scroll through months by using the two arrow links in the top left and right corners of the calendar.
The web part can be used for displaying documents of various types, but only one type can be displayed at the same time.



8.18.2 Using the web part

You can add the Event calendar web part to any web part zone on any page on your website. Go to **CMSDesk -> Edit**, switch to the page that you want to add the Event calendar to and switch to the **Design** tab. Click the **Add web part** icon of the target web part zone and from the web part list that appears, select **Events -> Event calendar** and click **OK**.

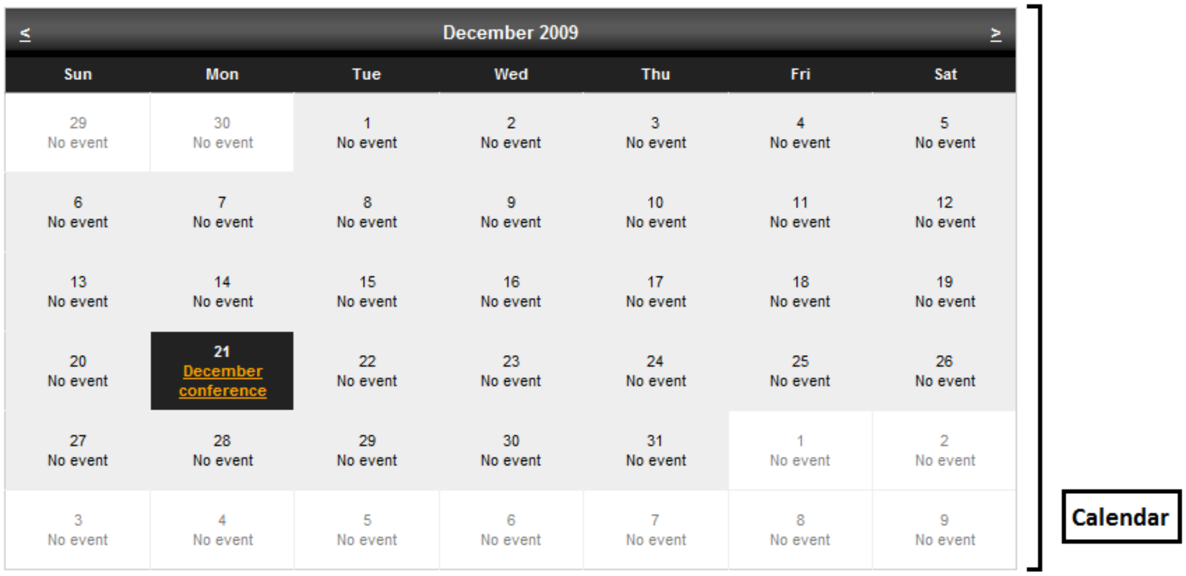
The web part properties dialog appears. Here is a list of properties specific for this web part:

Content	
Path	Path in the content tree to the displayed documents.
Day field	Field with the date that the document will be displayed on in the calendar.
Content filter	
Document types	Document types that will be displayed in the calendar. If there is a document in the location set in the **Path** parameter that is not of a type entered here, it will not be displayed in the calendar.
Combine with default culture	Indicates if the default language version of a document should be displayed if the document is not available in the current language.
Culture code	Culture code of the displayed versions of documents.
Maximum nesting level	Specifies the number of sub-levels in the content tree from that the documents will be displayed.
ORDER BY expression	The ORDER BY part of the SQL SELECT query used for selecting the documents.
Select only published	Indicates if only published documents should be displayed.
Site name	Code name of the website from that you want to display the content. If left empty, the current website is used.
WHERE condition	The WHERE part of the SQL SELECT query. If multiple links to the same document are displayed, this can be used to display only one of them.

Click **OK**. The Event calendar should be displayed in the selected web part zone and should be displaying documents from the selected location in the content tree.

8.18.3 Design and Styles

The design is controlled by CSS styles included in your CSS stylesheet for the given site. The following figure shows CSS classes used in the Event Calendar web part:



8.19 File import

8.19.1 Overview

The file import module allows you to import files including their entire folder structure from the disk to the Kentico CMS content repository (content tree). When files are uploaded this way, they are stored as **CMS.File** (or **CMS.Folder**) documents. The module allows large amounts of files to be selected, which eliminates the need to create documents and upload files in the content tree manually one-by-one.

The module's user interface can be found in **CMS Desk -> Tools -> File import**. The label at the top shows the path to the file import folder, which is where all files that you wish to import must be located. If the folder doesn't exist, you may need to create it on the disk. The default file import folder is **~/ CMSImportFiles**. You can define your custom file import path in **Site Manager -> Settings -> Files -> File import folder**.

For a more specific example of using this module to import files, please continue in the Importing files topic.

The Security topic shows how permissions can be set for this module. The use of this module should usually only be allowed for site administrators, as it doesn't check certain security settings from **Site Manager -> Settings -> Files**, such as allowed file extensions.

The file import module isn't the only way files can be uploaded to the Kentico CMS system. For other options and information about general file management, please refer to the Content management -> File Management chapter of this guide.

8.19.2 Importing files

The following is a step-by-step guide through the process of importing files:

1. Copy your files into the specified import folder on your local disk.

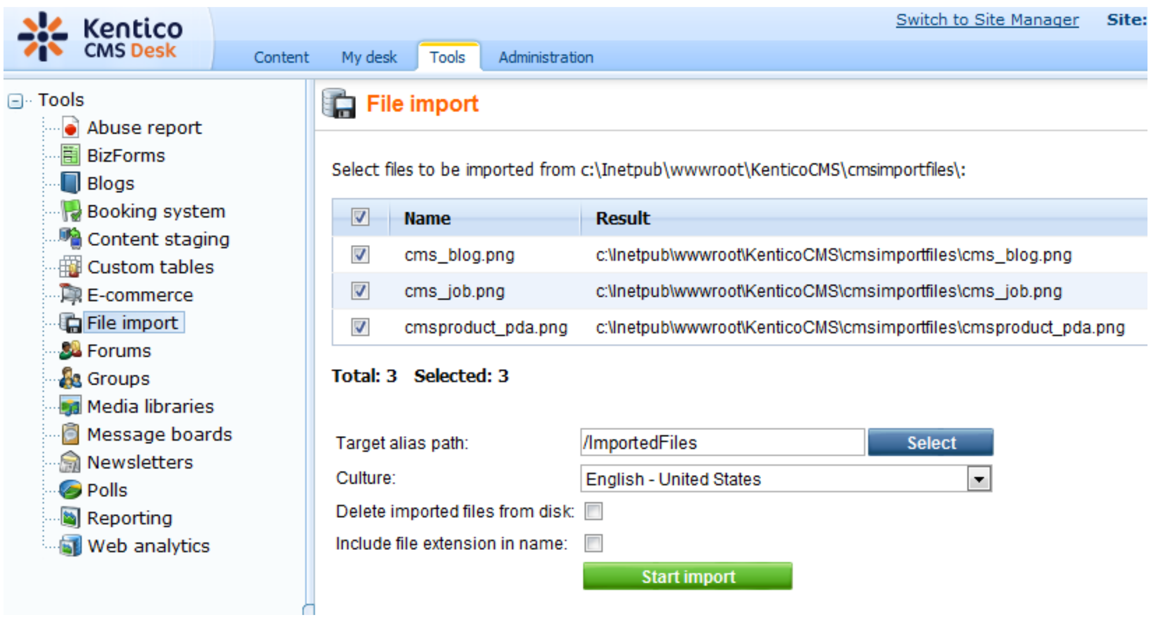
2. Go to **CMS Desk -> Tools -> File import**. You should see a list of files that are currently in the

import folder.

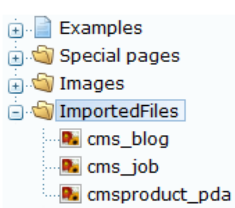
3. Choose which files should be imported using the checkbox next to each file and specify the following properties:

- **Target alias path**: path within the content tree where the files will be imported
- **Culture**: culture to which the uploaded files (documents) will be assigned
- **Delete imported files from disk**: if enabled, files in the import folder will be deleted after a successful import
- **Include file extension in name**: if enabled, the file extension is included in the *Document name* of the newly imported files

Click **Start Import**.



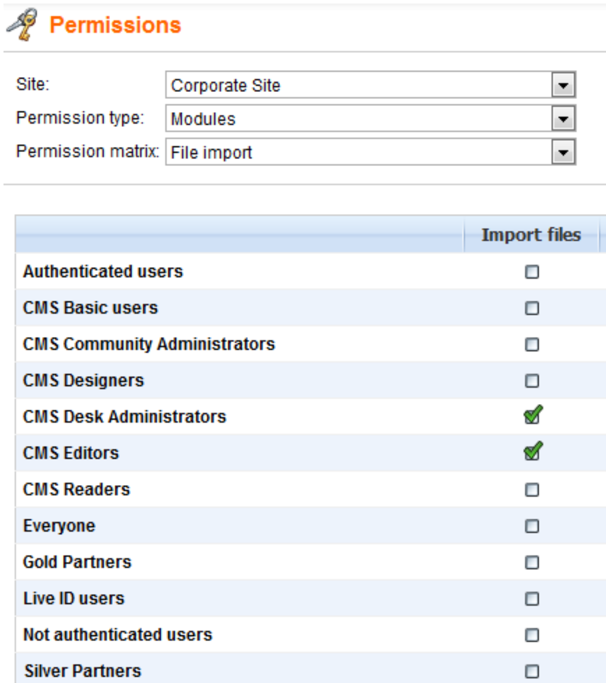
4. Now if you switch to the **Content** tab and locate the alias path that you specified in the previous step, you should see the files uploaded.



8.19.3 Security

Access rights to the file import module can be configured in **CMS Site Manager -> Administration -> Permissions**, after you select the **Modules -> File import** permission matrix. This can also be done in **CMS Desk -> Administration -> Permissions**, but only for the current site. The File import module only has one single permission:

- **Import files** - members of the specified roles are allowed to import files using the File import module

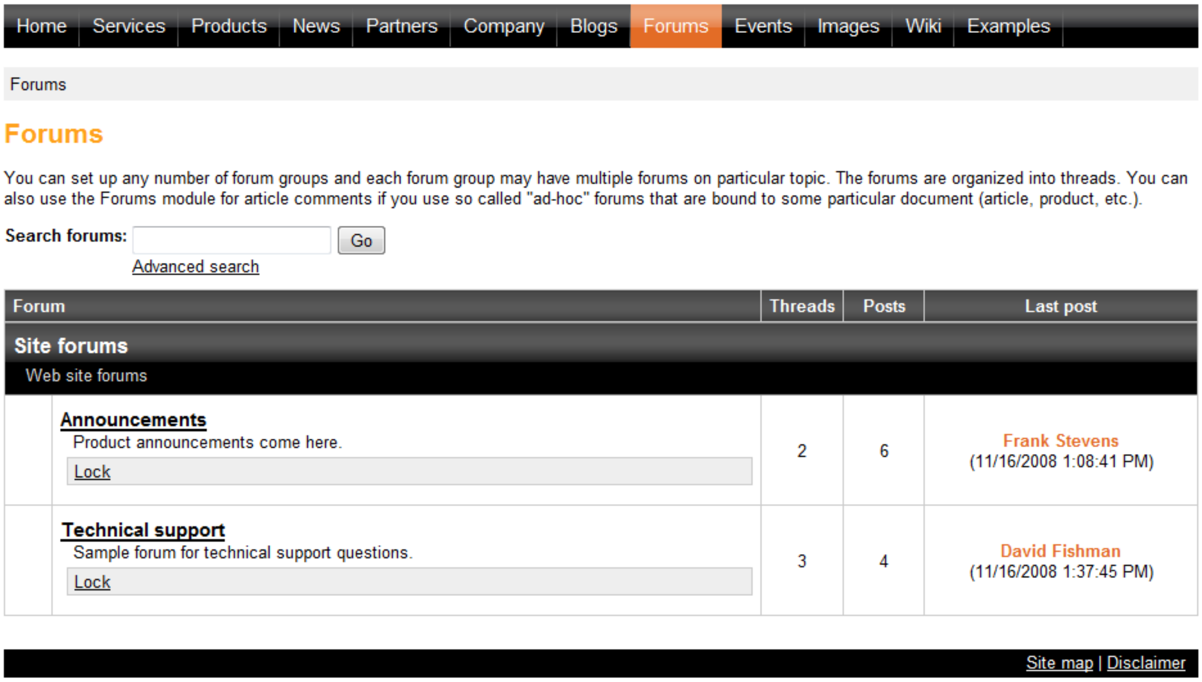


8.20 Forums

8.20.1 Overview

This module allows you to integrate full-featured forums into your website. The forums are highly configurable and allow you to:

- Organize forums into forum groups
- Open/lock forums
- Perform SQL or index-based full-text searches of forum content
- Manage all forum posts and threads of a particular forum
- Create moderated forums (posts needs to be approved before they are displayed on-line)
- Choose to require and/or display e-mails of the forum users
- Subscribe to receive notifications about all posts added to a forum or thread by e-mail
- Configure standard forum functions such as post attachments, BBCode support, adding posts to favorites, user avatars and more
- Set forums to use Friendly URLs
- Use your own custom forum layouts
- Enable a forum only for authenticated users
- Specify user roles that are allowed to use various forum functions via Security settings



There are two basic types of forums:

- Pre-defined forums - created by the administrator and then displayed on the website.
- Ad-hoc forums (article comments) - created for a single document when a visitor posts the first comment to the given document.

The Message boards module provides another option that allows users to post comments on your website. Please refer to the Modules -> Message boards chapter to learn more about it.

The Forums module can be managed in **CMS Desk -> Tools -> Forums**. Further settings can be found at **Site Manager -> Settings -> Forums** and are described in more detail in the Settings topic.

The Forums internals and API sub-chapter provides information about the database tables and classes used by the module and examples of how forums can be managed using the API.

Kentico CMS Community Site Guide contains some additional forum examples and tutorials:

- Part 1 -> Forums: Examples of the functionality and customization of groups.
- Part 2 -> Creating the Forums section: A step-by-step tutorial on how to create a sample forum section of a website.

You will need to have the sample Community Site installed to follow Kentico CMS Community Site Guide.

8.20.2 Creating a forum group

All forums are a part of some forum group. Pre-defined forums need to be created within some particular forum group and ad-hoc forums are automatically placed into the **AdHoc forum group** upon creation. A **forum group** usually contains forums related to the same topic. For example:

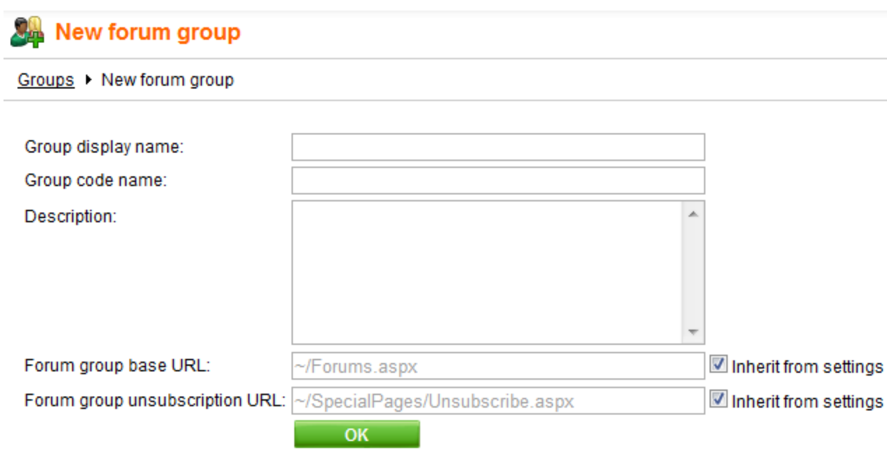
- Computers (forum group)
  - Announcements (forum)
  - Technical questions (forum)
  - FAQ's (forum)
- Web design (forum group)
  - CSS (forum)
  - XHTML (forum)

Creating a new forum group

Go to **CMS Desk -> Tools -> Forums** and click **New forum group**. Fill in the following fields:

- **Group display name:** the forum group name displayed on your website
- **Group code name:** the forum group name used in your code
- **Description:** description of the forum group displayed on your website
- **Forum group base URL:** URL displayed when the user accesses the forum; *eg. ~/MyForums.aspx*
- **Forum group unsubscription URL:** URL of a page where users can unsubscribe from the given forum group

Click **OK** to save.



Continue to the Creating a pre-defined forum topic to learn how to create individual forums within a forum group or the Adding an ad-hoc forum to the web topic to find out more about ad-hoc forums..

8.20.3 Creating a pre-defined forum

Pre-defined forums need to be created within a forum group before you can publish them on the website. See the Creating a forum group topic to learn how to create a forum group.

Creating a new forum

Go to **CMS Desk -> Tools -> Forums** and click **Edit** ( ) for some forum group. Go to the Forums tab and click **Add forum**. Enter the following details:

- **Forum display name** - the name of the forum displayed on your website
- **Forum code name** - the name of the forum used in your code
- **Description** - the description displayed on your website

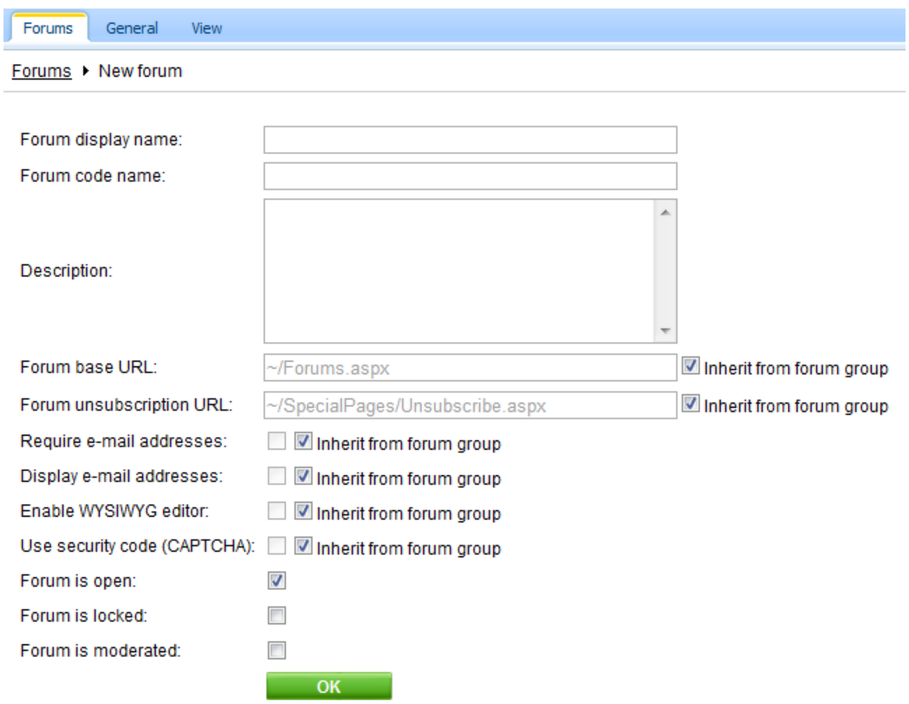
- **Forum base URL** - URL displayed when the user accesses the forum; *e.g. ~/MyForums.aspx*
- **Forum unsubscription URL** - URL of the page where users can unsubscribe from the given forum

- **Require e-mail address** - indicates if e-mail address should be required from the post author
- **Display e-mail addresses** - indicates if e-mail address of the post author should be displayed to other site visitors
- **Enable WYSIWYG editor** - indicates if the visitors can use the WYSIWYG editor for entering text
- **Use security code (CAPTCHA)** - indicates if the user needs to retype the security code displayed as an image - this feature helps you avoid spam in the forums

- **Forum is open** - indicates if the forum is visible and can be accessed
- **Forum is locked** - if checked, new posts can't be added to the forum, while the forum is still accessible for viewing
- **Forum is moderated** - indicates if the posts need to be approved by a forum moderator

The **Inherit from forum group** checkboxes displayed next to most properties are used to automatically load values from the settings of the parent forum group. To configure unique properties for a forum, simply clear the appropriate checkbox and enter the desired value.

Click **OK** to add the forum.



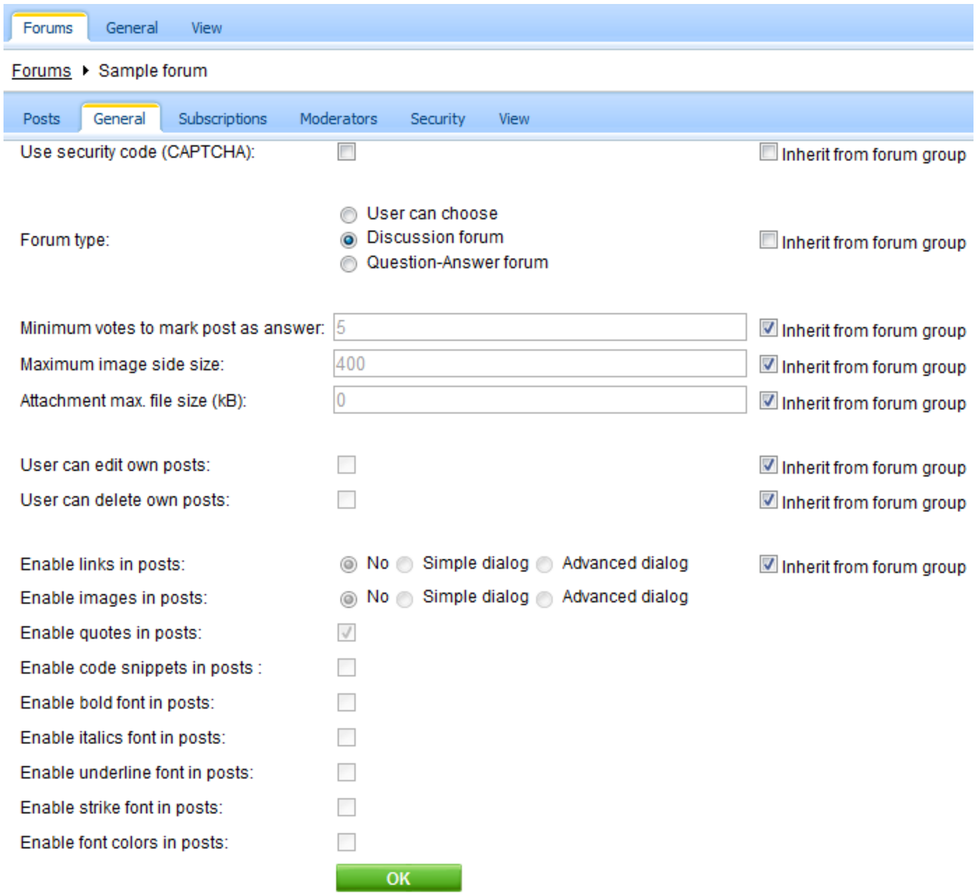
Now that the forum is created, it can be configured in more detail. Switch to its **General** tab, where you can edit the options that were entered when the forum was created, as well as the following additional properties:

- **Forum type** - determines the overall type of the forum. The following three types may be selected:
  - **User can choose** - when creating a new thread, users can choose from the following two types.
  - **Discussion forum** - news threads use a standard discussion format where users reply to previous posts.
  - **Question-Answer forum** - in forums of this type, the initial post of a thread is usually a question

and the replies are attempts to answer it. The forum includes a voting feature that allows users to mark individual replies as helpful answers. Once a post receives the amount of votes specified in the *Minimum votes to mark post as answer* property, it will be designated as a valid answer and displayed accordingly.

- **Minimum votes to mark post as answer -** determines the minimum amount of votes that a post must receive before it is marked as an answer in *Question-Answer* type forums.
- **Maximum image side size -** sets the maximum side size in pixels of images inserted into forum posts. If a larger picture is included in a forum post, it will be resized so that its larger side is equal to the entered value.
- **Attachment max. file size (kB) -** sets the maximum file size of forum post attachments in kB.
- **User can edit own posts -** indicates whether users are allowed to edit their own existing posts.
- **User can delete own posts -** indicates whether users are allowed to delete their own existing posts.

The properties at the bottom determine which BB code macros may be used by forum users to format the text of forum posts. Further details may be found in the BBCode support topic.



For information about the remaining tabs, please refer to the following topics:

- Managing forum posts
- Subscriptions
- Forum moderation
- Security

Once everything is configured as necessary, the forum can be placed on a website. The Publishing a [pre-defined forum on the website](#) topic describes how.

8.20.4 Publishing a pre-defined forum on the website

When you want to publish a forum on your website, you can use the built-in web parts in your page templates. You can find more details on each web part in [Kentico CMS Web Parts](#) reference.

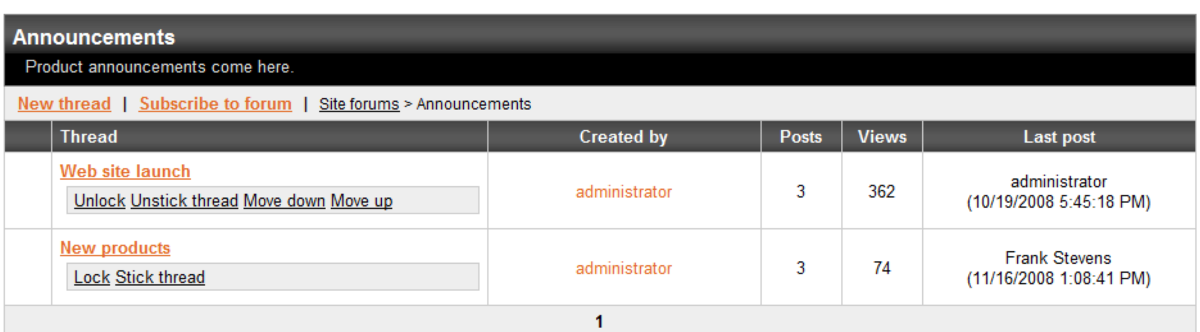
> **Web parts and ASPX page templates**
>
> If you're using ASPX page templates, you simply drag and drop the Forum web parts located in the CMSWebParts/Forums folder to your page and use them in a similar way.

Publishing a forum group on the website

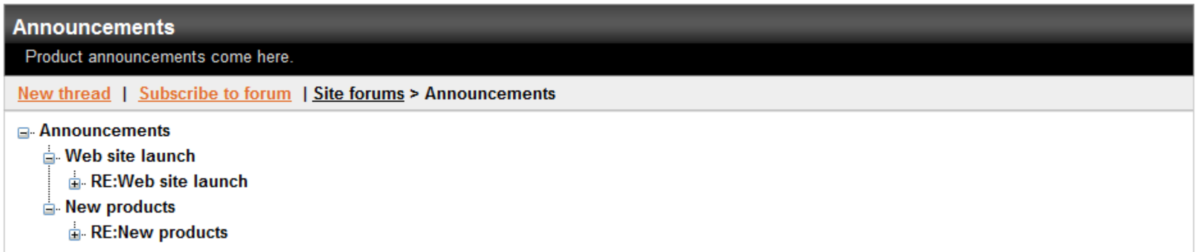
You can publish the whole forum group on the website using the **Forum group** (ForumGroup.ascx) web part. All you need to do is to select the appropriate **Group name** value in the web part properties. The default forum looks like this:

The forum threads in the selected forum can be displayed in two ways:

1. As a list of threads
- set the **Forum layout** property to **Flat**



2. As a tree of threads and posts
- set the **Forum layout** property to **Tree**



8.20.5 Adding an ad-hoc forum to the web

Ad-hoc forums are useful if you want to enable users to add comments to some page or article, but you do not want to create the forum for each document manually. You may also consider using Message boards for this purpose if you don't need a structured forum.

Ad hoc forums can be added using the **Forum (Single forum - General)** web part.

If you place the **Forum (Single forum - General)** web part to a page and set its **Forum name** property to *ad-hoc forum* (or *ad_hoc_forum* if you are using ASPX templates), the forum will be displayed on the website, but it will be actually created after some visitor adds a first post to the forum. After that, the forum will be created in the **AdHoc forum group** forum group in **CMS Desk -> Tools -> Forums**.

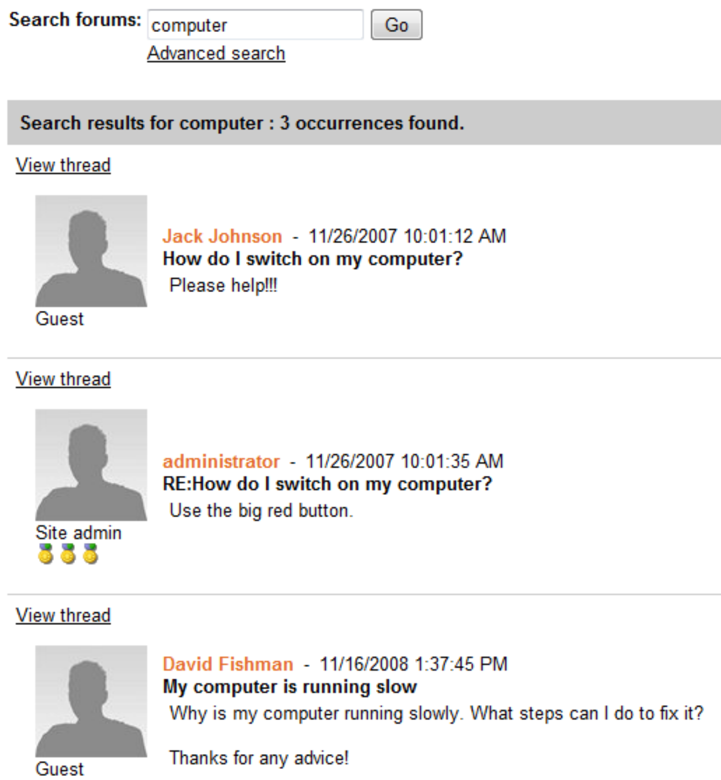
Ad-hoc forums are uniquely identified by the document they belong to.

The following two web parts are **deprecated**, and they are available only because of **backward compatibility** with versions prior to 4.0:

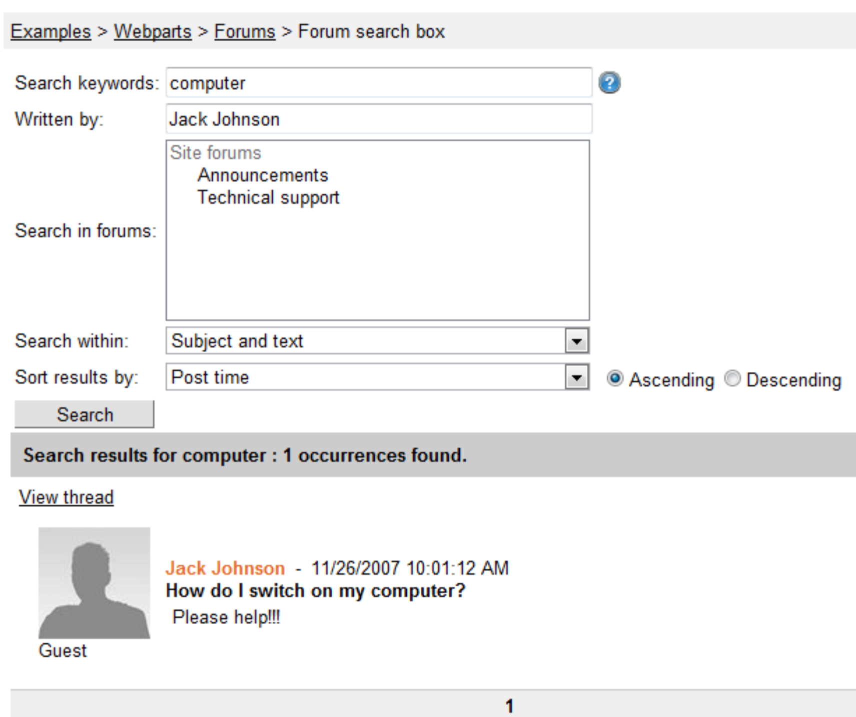
- **Forum (Single forum - Flat layout) -** this web part ensures backward compatibility with the former **Forum thread list** web part
- **Forum (Single forum - Tree layout)** - this web part ensures backward compatibility with the former **Forum tree** web part

8.20.6 Adding forum searching

If you want to let users search forum posts for some given text, you can use the **Forum search box** (ForumSearch.ascx) and **Forum search results** (ForumSearchResults.ascx) web parts. You will typically place them both on the same page. However, if you need to place the forum search box on a different page, you can set its **Redirect to URL** property to the page where you have the **Forum search results** web part, such as `~/SearchForum.aspx`.



You can also use the **Forum search - advanced dialog** (ForumExtendedSearchDialog.ascx) web part, which offers a larger dialog that allows users to specify multiple search parameters.



If you wish to use the basic search box by default, but want to let users have the option of using the advanced dialog as well, you can do so by entering the path to the page containing the **Forum search - advanced dialog** web part into the **Advanced search path** property of the **Forum search box** web part. The **Advanced search** link will then be displayed as seen in the first image.

These web parts use the SQL search engine.

If you prefer index-based searching, this can be provided by various Smart search web parts from the **Full-text search** category if they have a forums type search index set. Please refer to the Modules -> Smart search chapter for more details.

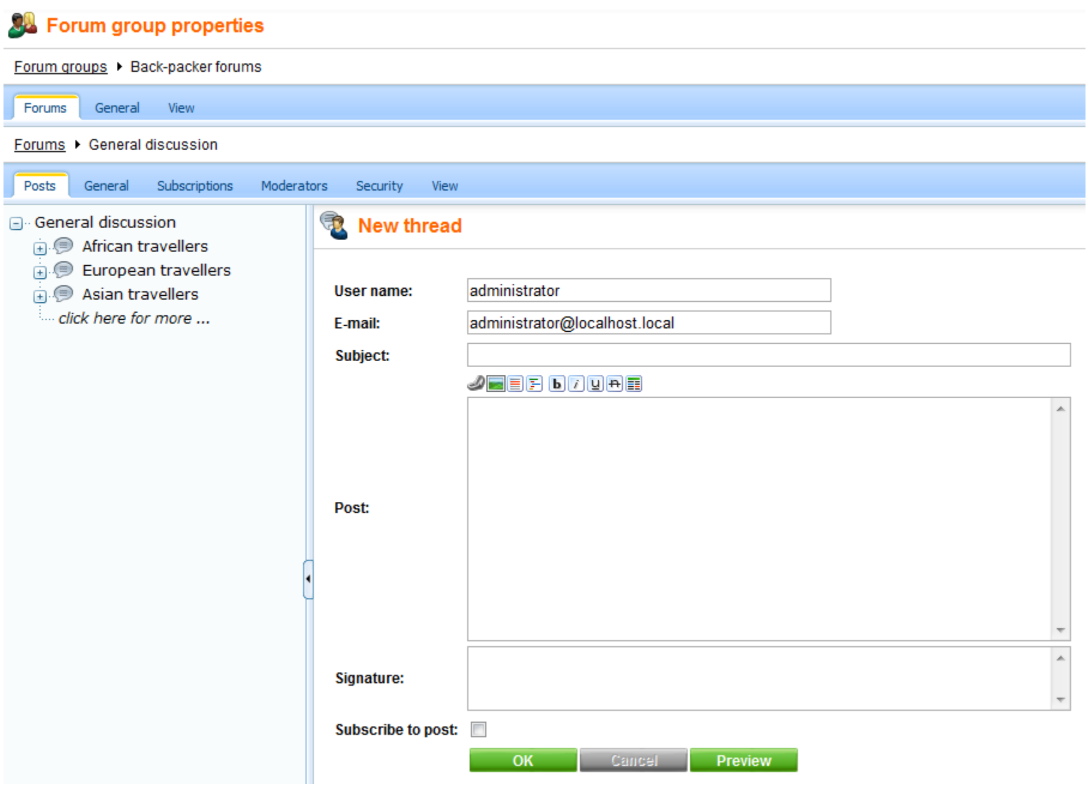
8.20.7 Managing forum posts

Threads and posts can be managed on the **Posts** tab when editing the forum that contains them.

Creating a new thread

When you navigate to the tab, you can initially see the **New thread** page, letting you add a new thread to the current forum group. The following details need to be filled in:

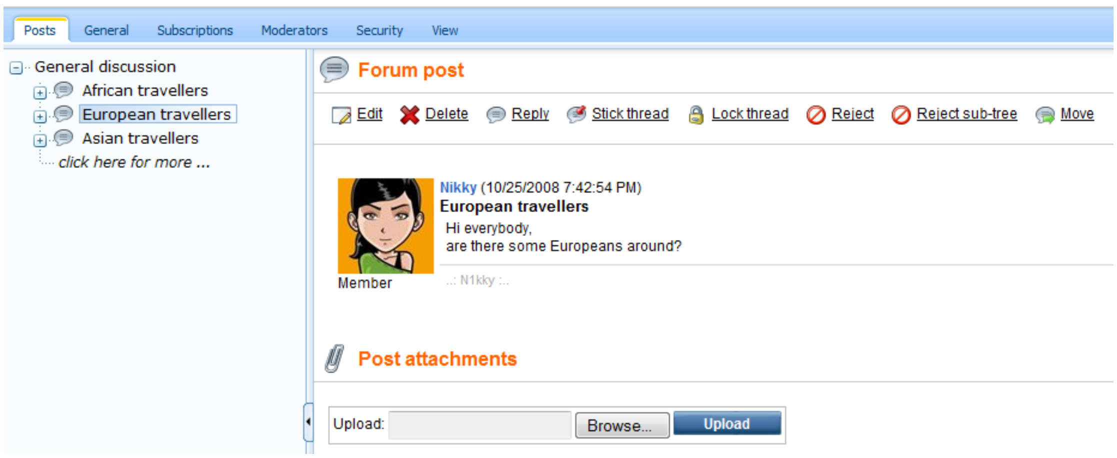
- **User name**: user name which will be used for the thread author
- **E-mail**: e-mail of the thread author
- **Subject**: subject of the initial post of the thread
- **Post**: text of the initial post of the thread
- **Signature**: signature under the initial post
- **Subscribe to post**: if enabled, notification e-mails about new posts in the thread will be sent to the e-mail address specified in the *E-mail* field



Managing particular posts

If you select a post from the tree on the left, you can perform the following actions:

- **Edit** - opens the post for editing - you can edit the post text, user's name, signature, etc.
- **Delete** - deletes the post
- **Reply** - opens a dialog using which you can send a reply to the post
- **Stick thread** - sticks this thread so that it will be displayed at the beginning of the forum
- **Split thread** - moves the post and its replies to a new thread
- **Lock thread** - locks the thread so that no more posts can be added to it
- **Reject** - rejects the post so that it is not displayed and needs approval to be displayed
- **Reject sub-tree** - performs the Reject action for the current posts and all its children
- **Move** - moves a thread to a new forum

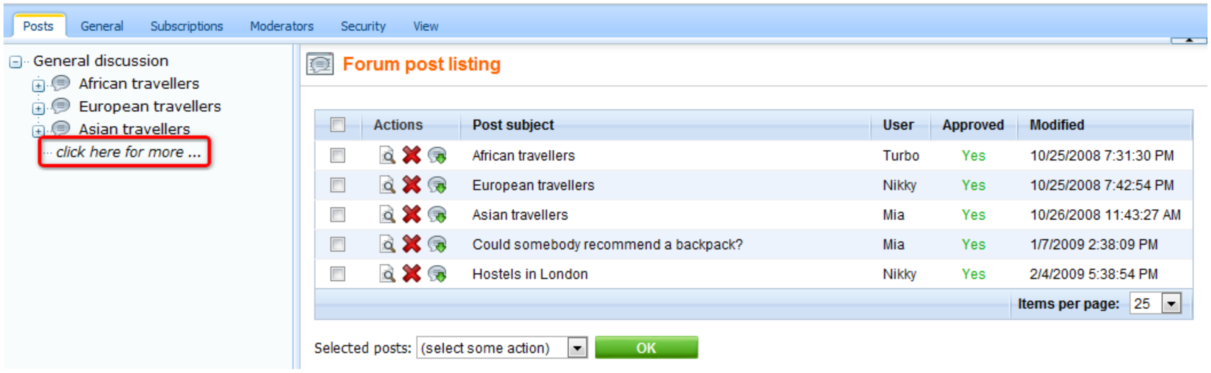


Forum post listing

In case that there are too many threads in the forum, you can limit the number of displayed threads using the **Max forum post nodes** settings key in **Site Manager -> Settings -> Forums**. If you use this key and there are more threads in the forum than the set value, the **Click here for more...** link is displayed at the bottom of the tree. After clicking this link, the **Forum post listing** page is displayed in the main area.

The **Forum posts listing** page initially displays all threads in the forum, while the following actions can be performed:

- **View** - displays the thread's initial post in the main area (this action is identical to clicking the post in the tree)
- **Delete** - deletes the whole thread including all child posts
- **Sub-posts** - displays sub-posts of the post in the grid in the main area
- **Parent post** - displays a list of sub-posts of the parent post in the main area (goes up one level)

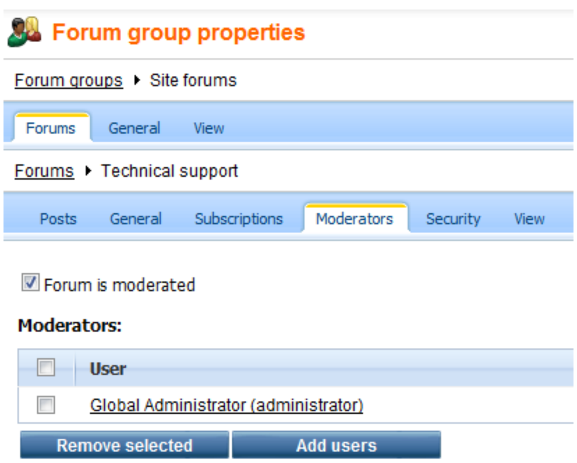


8.20.8 Forum moderation

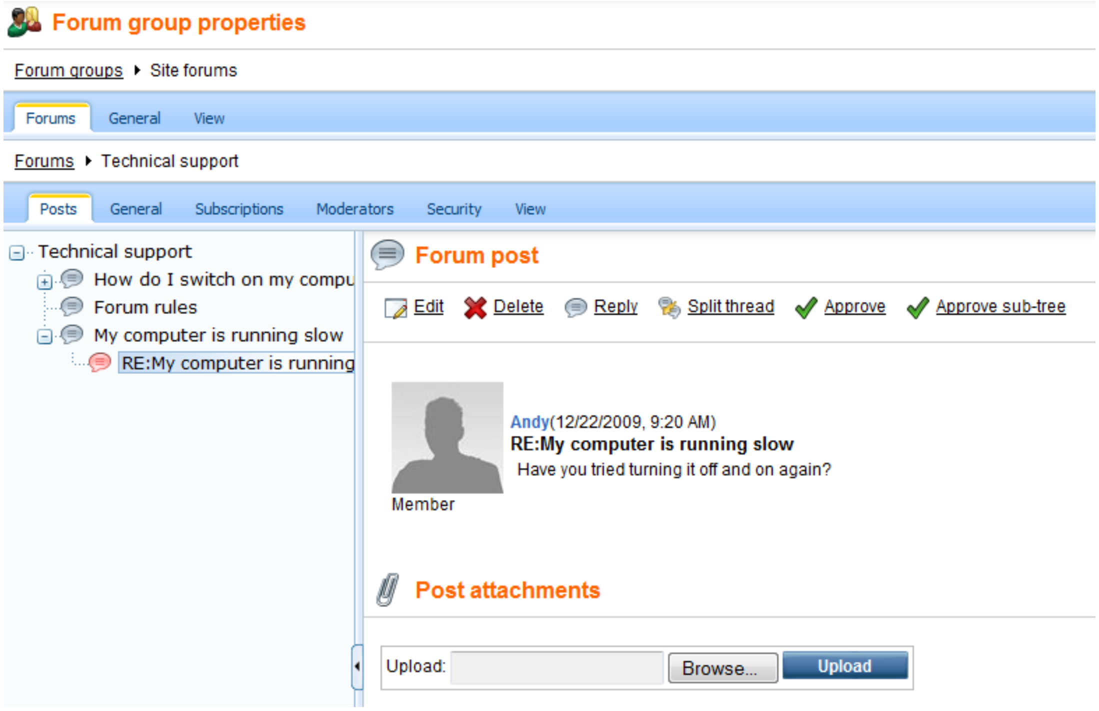
You can configure a forum so that it is moderated. It means that all posts must be approved by one of the forum moderators before they are published in the forum.

Configuring forum moderation

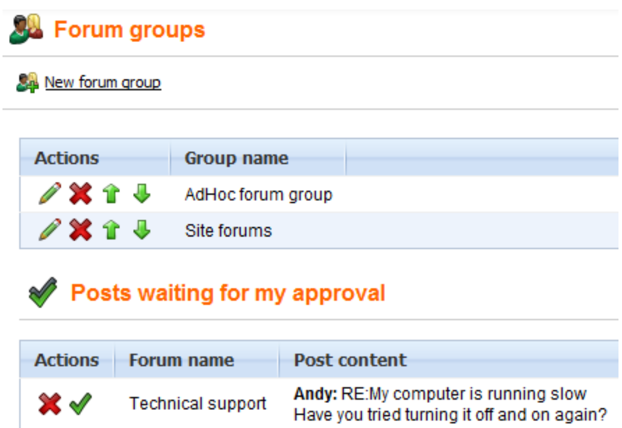
You can enable forum moderation and configure the list of moderators under the **Forum properties -> Moderators** tab. Even when this option is disabled, approval of posts might be required, e.g. when a Bad word is detected or when a post has been rejected.



If you're a moderator, you can moderate the forum posts either **directly on the website** or on the **Forum properties -> Posts** tab:



... or you can find the list of all posts waiting for your approval on the **Forum groups** dialog:



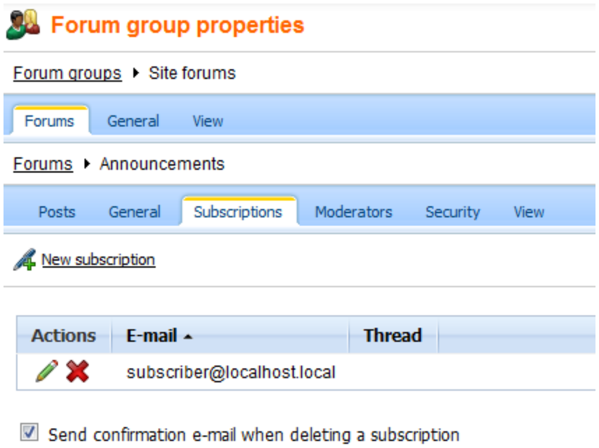
The moderators automatically receive an e-mail notification when a new item is waiting for approval. The e-mail template can be modified in **Site Manager -> Development -> E-mail templates -> Forums - Moderator notification**.

8.20.9 Subscriptions

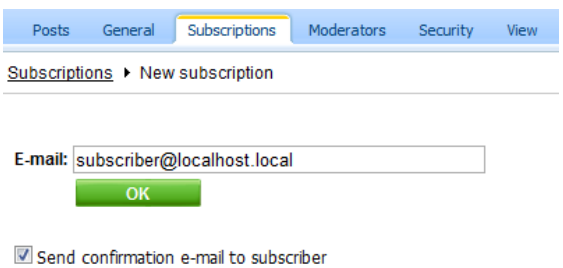
Subscriptions allow users to receive e-mail notifications when a new post is added to the selected forum.

There are several ways how a user can subscribe for receiving e-mail notifications:

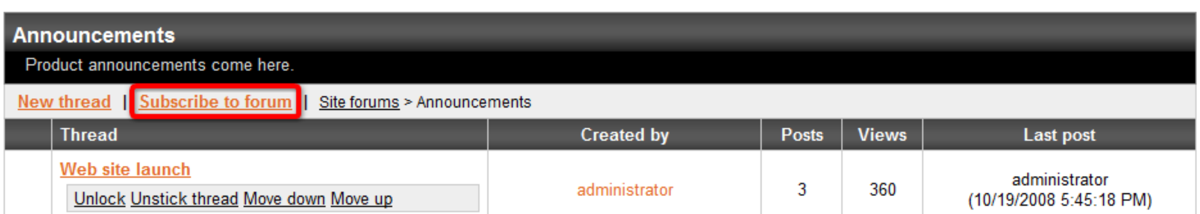
1. The forum administrator can subscribe users to the whole forum and manage all subscriptions at **Forum edit -> Subscriptions**.



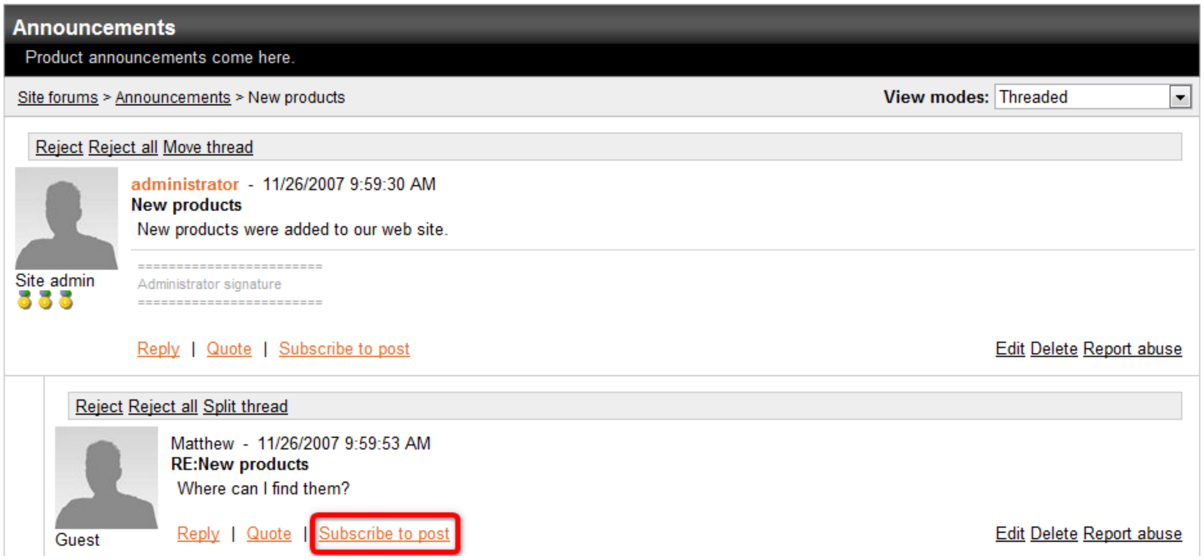
After clicking the **New subscription** link, the following dialog depicted in the screenshot below will be displayed. You only need to enter the user's e-mail address and click **OK**. If you do it with the **Send confirmation e-mail to subscriber** check-box checked, the user will receive a confirmation e-mail, telling them that they have been subscribed. These confirmation e-mails are sent automatically when the user subscribes on-site, as described in the following two points.



2. The user clicks the **subscribe to forum** link at the top of the forum:



3. The user subscribes to a particular post and sub-posts:



The e-mails are sent to the subscribed users based on the **Forums - New post** e-mail template that can be edited in the **Site Manager -> Development -> E-mail templates** dialog.

The user can unsubscribe using the link in the e-mail.

Forum base URL and forum e-mails

It's important to set a correct **Forum base URL value** in the forum properties so that the notification e-mails contain valid links to the forum. Use URL in format http://localhost/kenticocms/forums.aspx **- enter the full URL of the page with the forum, including the domain.**

8.20.10 Forum post attachments

Users can attach files to forum posts. To enable users to do this, you have to assign the **Attach files** permission on the forum's **Security** tab to the desired user roles or all authenticated users. See the Security topic for more details. There are also some additional settings related to attachments:

In **Site Manager -> Settings -> Forums**, you can make the following setting:

- **Forum attachments allowed extensions** - you can specify extensions of files that can be attached to forum posts; extensions should be entered without dots and separated by semicolons; if blank, all extensions are allowed

In web part properties of the **Forum group** web part, you can make the following settings:

- **Display attachment image** - if checked, attached images will be displayed as images, not only as links
- **Attachment image maximal side size** - if an attached image is larger than the entered value, it will be resized so that its larger side's size is equal to the entered value

When editing a forum in **CMS Desk -> Tools -> Forums**, you can set the following property on the

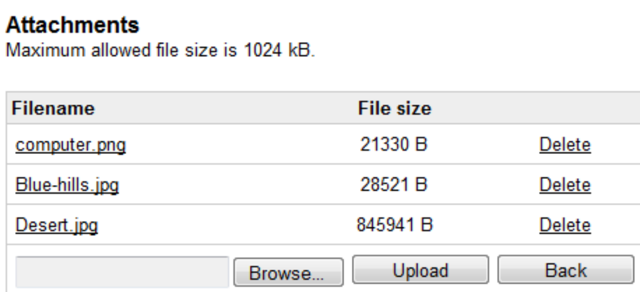
General tab:

- **Attachment max. file size (kB) -** you can define the maximal size of an attached file in kB

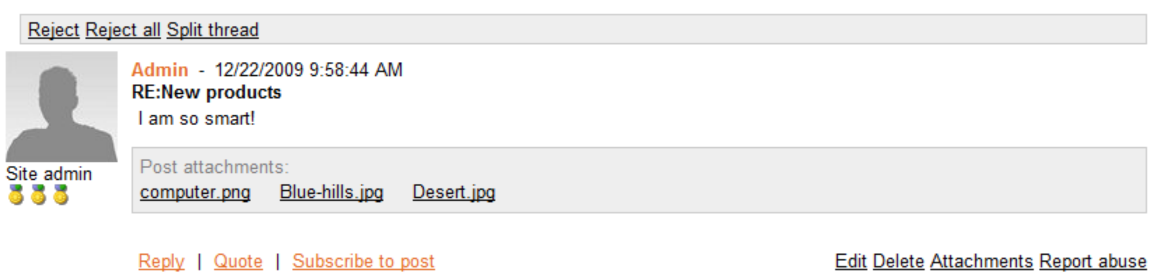
Adding attachments to a forum post

When adding a post to a forum, users can check the **Attach file(s)** check box at the bottom of the post adding dialog.

After clicking **OK**, a new dialog will be displayed, where users can upload the attachments. When all desired images are uploaded, users can click the **Back** button, which will take them back to the forum.



Back in the forum thread, links to uploaded attachments will be displayed with the post, as you can see in the screenshot below.



8.20.11 BBCode support

Bulletin Board code (the *BBCode* abbreviation is commonly used) is a lightweight markup language designed to let users format text of their messages. It is used in many forums on the web, not just on websites crated with Kentico CMS. Its tags are similar to HTML tags and are entered in square brackets.

Users can use BBCode in their forum posts only if it is enabled. You have to allow use of BBCode when editing ( ) a forum or forum group, on the **General** tab of its editing interface in **CMS Desk -> Tools ->**

Forums.

The following table explains particular BBCode tags. The first column shows icons on the BB editor toolbar. Clicking one of these icons encloses text selected in the text within the respective tags. The second column shows examples of text enclosed within the tags. The **Description** column explains what effects the tags have on the formatting. The last column lists respective properties on the **General** tab that have to be enabled in order for the tags to be functional.

Icon	Example	Description	Property on the General tab
	[url]http://example.org[/url] [url=http://example.com]Example[/url]	Makes the text a link leading to the URL. One of the following three options can be selected: **No** - the macros are not functional and the icon is not displayed **Simple dialog** - clicking the icon displays a simple dialog where the URL of the link can be entered manually **Advanced dialog** - clicking the icon displays the Insert link dialog where the URL can be selected in various ways	Enable links in posts
	[img]http://www.imagesxy.com/img.bmp[/img] [img=200x50]http://www.imagesxy.com/img.bmp[/img] [img=200]http://www.imagesxy.com/img.bmp[/img]	Displays an image located at the URL. The optional parameter resizes the image. It can be added either in format *<width>x<height>* or *<max side size>*. One of the following three options can be selected: **No** - the macros are not functional and the icon is not displayed **Simple dialog** - clicking the icon displays a simple dialog where the URL of the link can be entered manually **Advanced dialog** - clicking the icon displays the Insert image dialog where the image can be selected in various ways	Enable image macros in posts
	[quote]quoted text[/quote] [quote=Administrator]quoted text[/quote]	Displays text in a grey box; used for quotations. The optional parameter displays *Administrator wrote:* and the quoted text on a new line.	Enable quote macros in posts
	[code]code example[/code]	Displays text in monospaced format;	Enable code snippet

		used for code snippets.	macros in posts
	[b]bold text[/b] [strong]bold text[/strong]	Makes the text bold.	Enable bold font macros in posts
	[i]italicized text[/i] [em]italicized text[/em]	Makes the text italic.	Enable italics font macros in posts
	[u]underlined text[/u]	Underlines the text.	Enable underline font macros in posts
	[s]strikethrough text[/s]	Strikes the text through.	Enable strike font macros in posts
	[color=red]Red Text[/color] [color=#f00]Red Text[/color] [color=f00]Red Text[/color] [color=#ff0000]Red Text[/color] [color=ff0000]Red Text[/color]	Sets the text color.	Enable font color macros in posts

URLs in BBCode macros

All URLs in macros (URL, IMG) are validated as a URL to avoid XSS and resolved into their absolute URL equivalents. The following URL formats can be used:

- **www.google.com** – URL starting with www.
- **http://devnet.kentico.com** – URL starting with protocol
- **~/CMSDesk/default.aspx** – Virtual path
- **../default.aspx** – Relative URLs
- **/KenticoCMS/default.aspx** – Server relative URL

API for BBcode macros

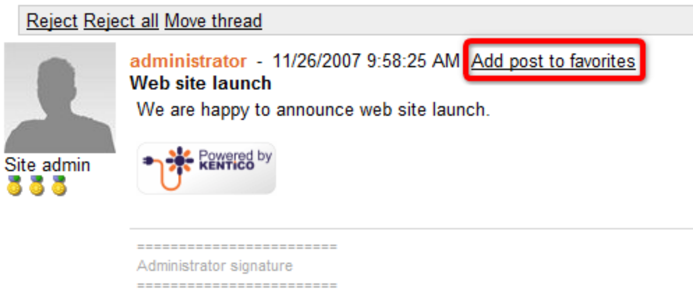
There is an easy way to resolve macros in ASPX or codebehind code. To resolve all the macros (recommended), use the following method:

string **CMS.CMSHelper.CMSContext.ResolveDiscussionMacros**(string **inputText**)

Or you can use the **CMS.GlobalHelper.DiscussionMacroHelper** class to resolve macros with particular settings using the **ResolveMacros** method from an object of this class.

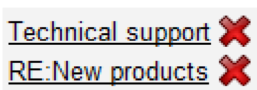
8.20.12 Forum favorites

If you enable the **Enable favorites** web part property of the **Forum group** web part, forum posts will be displayed with the **Add post to favorites** link, as you can see in the screenshot below.



If a user clicks this link, the post will be added to their favorite posts.

If you place the **Forum favorites** web part to some page, the web part will display links to favorite posts of the current user. You can see the appearance of the web part with two favorite posts in the screenshot below.



8.20.13 Friendly URLs

Forums display content based on the current values of the **forumid** and **threadid** query string parameters. By default, the URL of a forum thread might look like the following:

- **<domain>/Forums.aspx?forumid=3&threadid=12**

A friendly URL of the same forum thread looks like the following:

- **<domain>/Forums/f4/t13/Frequently-asked-questions.aspx**

Having your forum URLs in this format is a good practice when it comes to SEO (Search Engine Optimization).

Enabling friendly URLs

For this to be enabled, you have to do the following two tasks:

1. Set the following properties of the **Forum group** web part:

- **Use friendly URLs** - check to enable the Friendly URLs feature
- **Friendly base URL** - enter the base part of the URL, which is displayed after the domain name; e.g. **/Forums** for the example above and for the document aliases listed below
- **URL extension** - extension that will be used at the end of the friendly URL;

2. Assign the following document aliases to the document containing the Forum group web part. The **/Forums** part of each alias must be equal to the value in the **Friendly base URL** property of the Forum group web part:

- **/Forums/f{forumid}/{whatever}**
- **/Forums/f{forumid}/fp{fpage}/{whatever}**

- **/Forums/f{forumid}/t{threadid}/{whatever}**
- **/Forums/f{forumid}/fp{fpage}/t{threadid}/{whatever}**
- **/Forums/f{forumid}/t{threadid}/tp{tpage}/{whatever}**
- **/Forums/f{forumid}/fp{fpage}/t{threadid}/tp{tpage}/{whatever}**

Single forum friendly URLs

If you want to enable friendly URLs for a single forum, the **Forum (Single forum - General)** web part should be set the same way as described in step 1, but only the following two document aliases need to be added to the document:

- **/Forums/t{threadid}/{whatever}**
- **/Forums/t{threadid}/tp{tpage}/{whatever}**

8.20.14 Customizing forum design

Forum layouts are stored in **<web project>\CMSModules\Forums\Controls\Layouts**. The folder contains three sub-folders by default.

Name	Ext	Size
[..]		<DIR>
[Custom]		<DIR>
[Flat]		<DIR>
[Tree]		<DIR>

The **Flat** and **Tree** folders contain controls for the identically named layouts. The **Custom** folder can be used for defining custom layouts.

To **create a custom layout**, you need to create a sub-folder inside the **Custom** folder. The name of this sub-folder will be used as the name of the layout. The sub-folder has to contain controls with the same names as included in the Flat or Tree folders. You can see a screenshot of these files below. When writing the controls' code, make sure that they inherit from **ForumViewer**.

Name	Ext
[..]	
Attachments	ascx
Attachments.ascx	cs
Forums	ascx
Forums.ascx	cs
SearchResults	ascx
SearchResults.ascx	cs
SubscriptionEdit	ascx
SubscriptionEdit.ascx	cs
Thread	ascx
Thread.ascx	cs
ThreadEdit	ascx
ThreadEdit.ascx	cs
Threads	ascx
Threads.ascx	cs

8.20.15 Security

The security model of the Forums module has two parts:

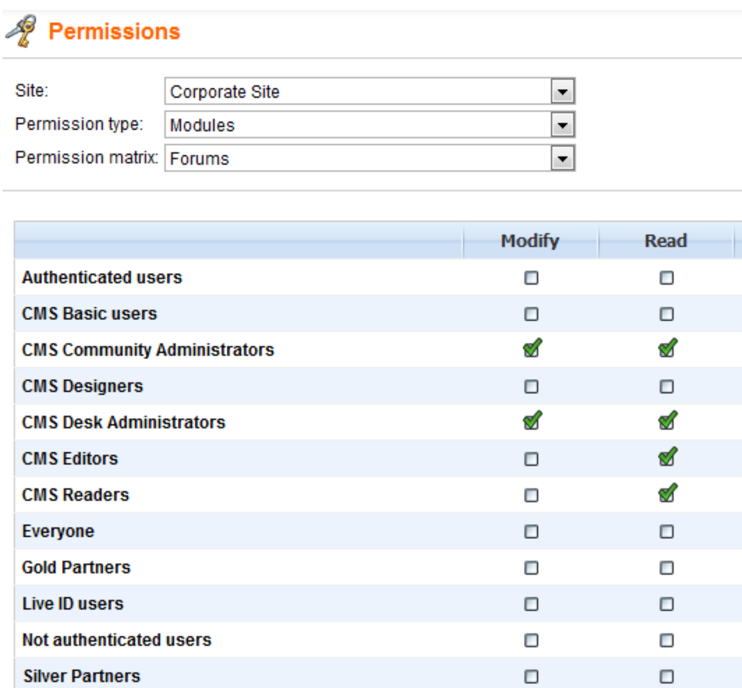
1. Security of the Forums module administration interface.
2. Security of the forums published on the website.

Forums module administration interface

Access to the Forums module administration interface in **CMS Desk -> Tools** can be managed in the **Modules -> Forums** permission matrix at **CMS Site Manager -> Administration -> Permissions** (or **CMS Desk -> Administration -> Permissions**; only applies to the current site):

- **Modify** - allows users to modify forum settings
- **Read** - allows users to only read forum settings

Users without any permissions who are moderators of at least one forum are allowed to access the **Posts waiting for my approval** dialog only.



Security of forums published on a website

If you **Edit** () some forum and switch to its **Security** tab, the permission matrix displayed in the screenshot below will be displayed.

Columns of the matrix represent the following actions:

- **Access to forum** - defines who can enter the forum and view posts
- **Attach files** - defines who can attach files to forum posts
- **Mark as answer** - defines who can mark posts as answers in Question - Answer forums
- **Post** - defines who can add posts to the forum

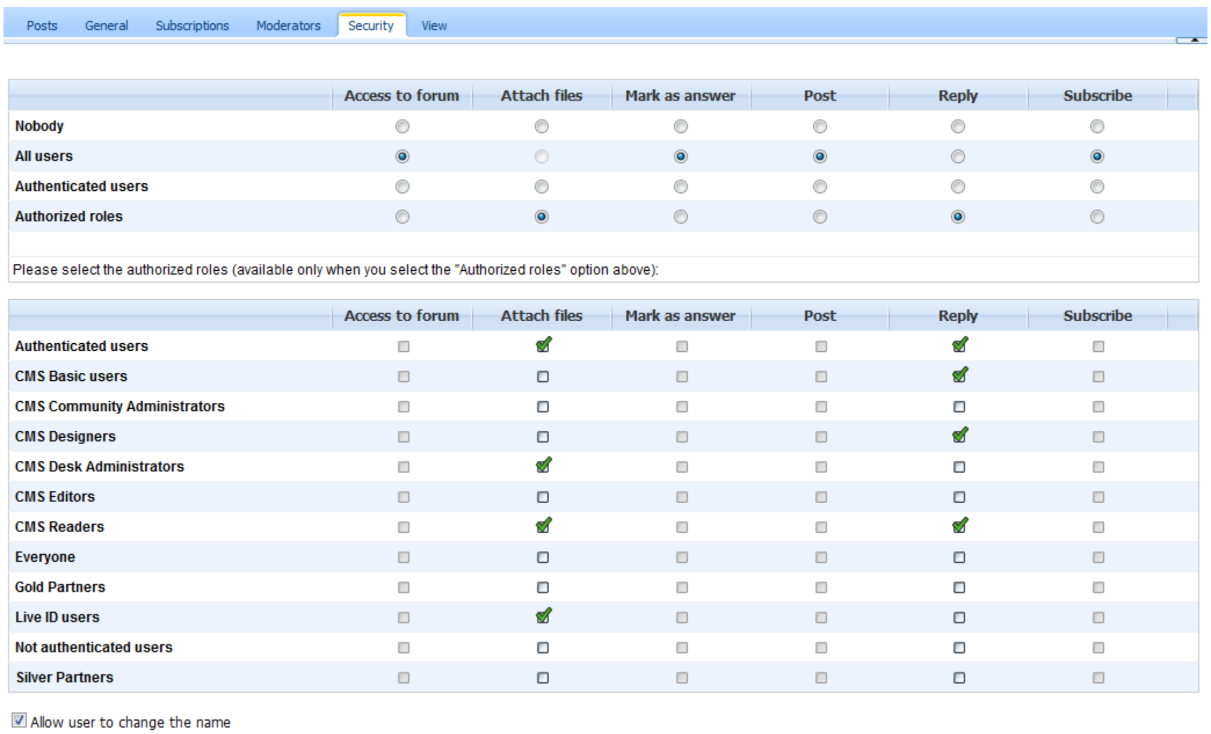
- **Reply** - defines who can reply to forum posts
- **Subscribe** - defines who can subscribe for receiving notifications about new posts in the forum

Rows in the top part of the matrix have the following meanings:

- **Nobody** - the action can not be performed by anyone
- **All users** - anybody can perform the action
- **Authenticated users** - only signed-in registered users can perform the action
- **Authorized roles** - only members of roles specified in the lower part of the matrix can perform the action

Below the permission matrix, there is one more check-box:

Allow user to change the name - if checked, users can change their name displayed with a forum post when entering the post; if unchecked, their user name will be used



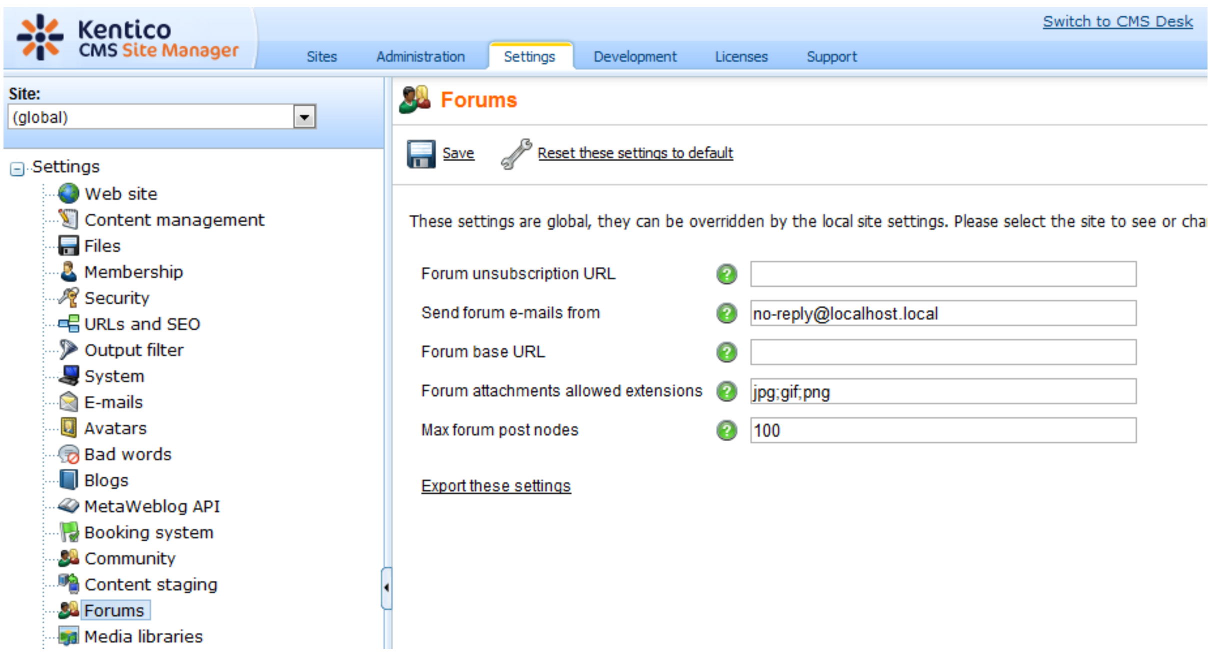
The following properties of the **Forum group** web part are also related to forum security:

- **Hide forum to unauthorized users** - indicates whether forums for which the user has no permissions are visible for them in the list of forums in a forum group
- **Redirect unauthorized users** - determines whether to redirect unauthorized users to the logon page or whether to display only an info message
- **Access denied page URL** - URL where the user is redirected when trying to access forum for which they are not authorized

8.20.16 Settings

Settings of the Forums module are located in **Site Manager -> Settings -> Forums**. The following settings are available:

- **Forum unsubscription URL** - URL that will be used for unsubscriptions from receiving notifications about new forum posts. Can be inherited by forum groups and further by particular forums. The **Forum unsubscription** web part must be placed on the page for the unsubscriptions to work correctly.
- **Send forum e-mails from** - e-mail address that will be used as the sender address of forum notification e-mails.
- **Forum base URL** - URL that will be displayed when viewing a forum and in forum notification e-mails. e.g. *~/Forum.aspx*
- **Forum attachments allowed extensions** - you can specify extensions of files that can be attached to forum posts; extensions should be entered without dots and separated by semicolons; if blank, all extensions are allowed
- **Max forum post nodes** - maximum number of forum post nodes displayed in the forum post tree when editing a forum in CMS Desk -> Tools -> Forums; when more nodes are in the forum group than this value, the *click here for more ...* link will be displayed at the bottom of the tree, letting you display a list of all nodes in the main area



8.20.17 Forums internals and API

8.20.17.1 Database tables and API classes

The Forums module uses the following database tables:

- **Forums_Forum** - contains forums and their settings
- **Forums_ForumGroup** - contains forum groups and their settings
- **Forums_ForumModerators** - contains information about forum moderators
- **Forums_ForumPost** - contains forum posts and information about them
- **Forums_ForumRoles** - contains information about forum permissions
- **Forums_ForumSubscription** - contains information about forum subscriptions
- **Forums_Attachment** - contains forum post attachments and information about them
- **Forums_UserFavorites** - contains favorite posts of users

The Forums API is provided by the following **CMS.Forums** namespace classes:

- **ForumInfo**, **ForumInfoProvider -** these classes provide functionality for managing individual forums
- **ForumGroupInfo**, **ForumGroupInfoProvider -** these classes provide functionality for managing forum groups
- **ForumPostInfo**, **ForumPostInfoProvider -** these classes provide functionality for managing forum posts and threads

The following topics show examples of how these classes can be used:

- Creating forums
- Creating threads

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.20.17.2 Creating forums

The following sample code shows how you can create a new forum group and forum and add them to the system:

[C#]

```
using CMS.Forums;
using CMS.CMSHelper;
using CMS.SiteProvider;

...

        // Create new forum group info
        ForumGroupInfo fgi = new ForumGroupInfo();

        // Set properties
        fgi.GroupName = "CustomForumGroup";
        fgi.GroupDisplayName = "Custom forum group";
        fgi.GroupSiteID = CMSContext.CurrentSiteID;
        fgi.GroupAuthorDelete = true;
        fgi.GroupAuthorEdit = true;
        fgi.GroupDisplayEmails = true;

        // Save object to database
        ForumGroupInfoProvider.SetForumGroupInfo(fgi);



        // Get Forum group object
        fgi = ForumGroupInfoProvider.GetForumGroupInfo("CustomForumGroup", CMSContex
```

```
        // If forum group exists
        if (fgi != null)
        {
            // Create new forum object
            ForumInfo fi = new ForumInfo();

            // Set properties
            fi.ForumName = "CustomForum";
            fi.ForumDisplayName = "Custom forum";
            fi.AllowAccess = SecurityAccessEnum.AllUsers;
            fi.AllowAttachFiles = SecurityAccessEnum.AuthenticatedUsers;
            fi.AllowPost = SecurityAccessEnum.AllUsers;
            fi.AllowReply = SecurityAccessEnum.AllUsers;
            fi.AllowSubscribe = SecurityAccessEnum.AllUsers;
            fi.ForumGroupID = fgi.GroupID;
            fi.ForumOpen = true;
            fi.ForumModerated = false;
            fi.ForumThreads = 0;
            fi.ForumPosts = 0;
            fi.ForumSiteID = fgi.GroupSiteID;

            // Save new forum to database
            ForumInfoProvider.SetForumInfo(fi);

        }
```

8.20.17.3 Creating threads

The following sample code shows how you can create a new thread and add it to a forum:

[C#]

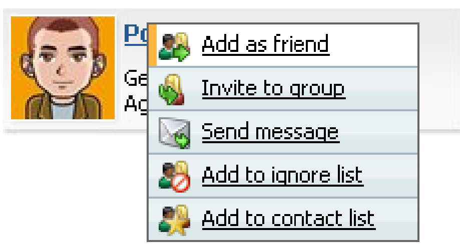
```
using System;
using CMS.Forums;
using CMS.CMSHelper;

...

        // Get forum object
        ForumInfo fi = ForumInfoProvider.GetForumInfo("CustomForum", CMSContext.Curr

        // If forum exists
        if (fi != null)
        {
            // Create new thread object
            ForumPostInfo fpi = new ForumPostInfo();

            // Set properties
            fpi.PostUserID = CMSContext.CurrentUser.UserID;
            fpi.PostUserMail = CMSContext.CurrentUser.Email;
            fpi.PostUserName = CMSContext.CurrentUser.UserName;
            fpi.PostForumID = fi.ForumID;
```

```
        fpi.PostTime = DateTime.Now;
        fpi.PostApproved = true;
        fpi.PostText = "This is a forum post";
        fpi.PostSubject = "Post subject";

        // Save thread to database
        ForumPostInfoProvider.SetForumPostInfo(fpi);

    }
```

8.21 Friends

8.21.1 Overview

The Friends module is one of the social-networking features of Kentico CMS. It allows you to add the Facebook-style friendship functionality to your site. The module enables registered website users to request another user's friendship. When the friendship gets approved by the second user, the two users become friends. The process of requesting a friendship is described in the Requesting friendships topic.



Management of friendships can be performed either by the users themselves, or by the site administrators. Both options are described in the Friends management topic. Users can manage only their own friendships, while users with appropriate permissions (described in the Security topic) can manage friendships of all users in the system.

The module comes with several web parts. Using these, you can add the friendship management functionality to the live site. All web parts related to the module are listed in the Available web parts topic.

By default, becoming friends brings just some improvements to the way the users can communicate and publish details about themselves, as described in the Examples of use topic. However, the friendship functionality may be leveraged largely in your custom code, enabling you to use these relationships between users for your specific purposes.

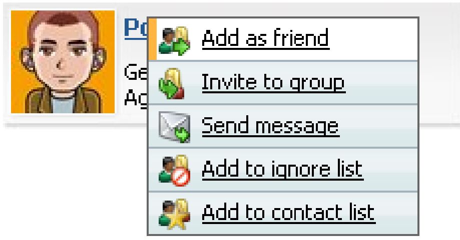
8.21.2 Requesting friendships

How to make the request?

There are several ways how users can request other users' friendship:

Context menus

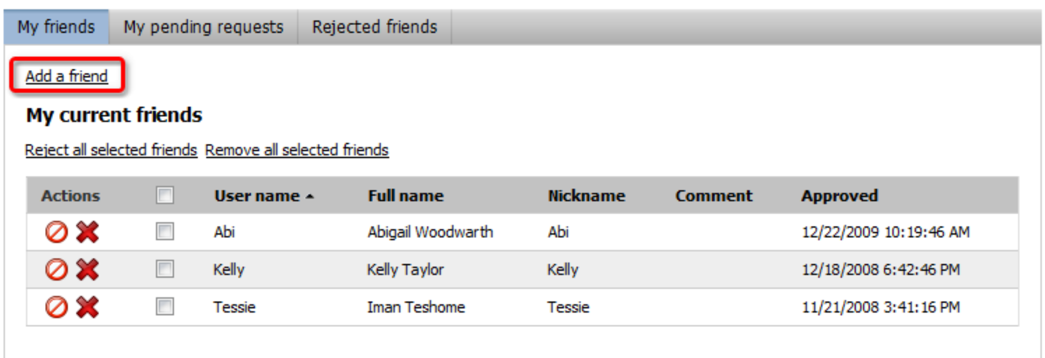
As described in Development -> Document types and transformations -> Context menus in transformations, the **Users viewer** web part may be equipped with a context menu when an appropriately written transformation is used. This context menu contains the **Add as friend** option, enabling registered website users to request this user's friendship.



Request friendship and My friends web parts

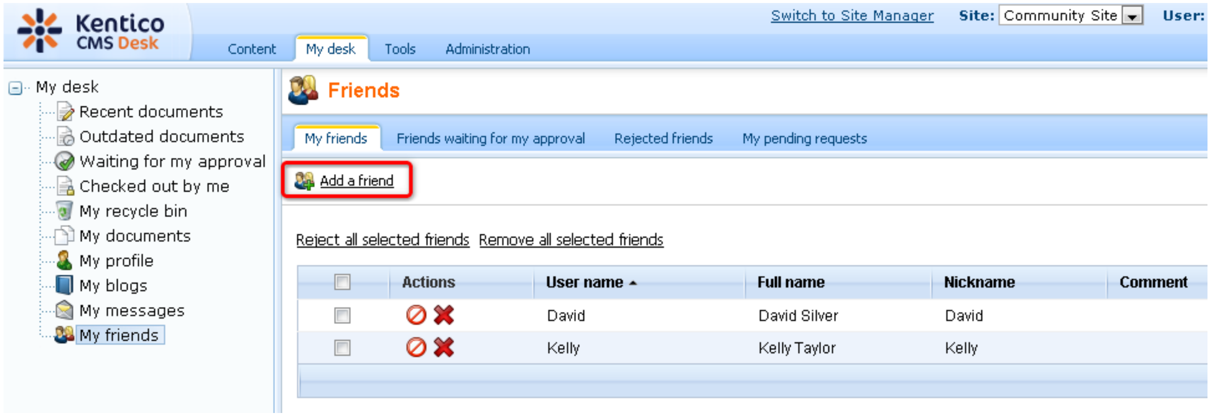
The **Request friendship** web part is just a tiny link with customizable text. Clicking it opens the **Add a new friend** pop-up dialog.

Requesting friendships on the live site is also possible via the **My friends** web part. It contains the **Add a friend** link, which opens the same **Add a new friend** dialog mentioned above.



CMS Desk -> My Desk -> My friends

Users with access to **CMS Desk** can request friendships from within **My desk -> My friends**. Here again, they can find the **Add a friend** link, which opens the **Add a new friend** dialog.

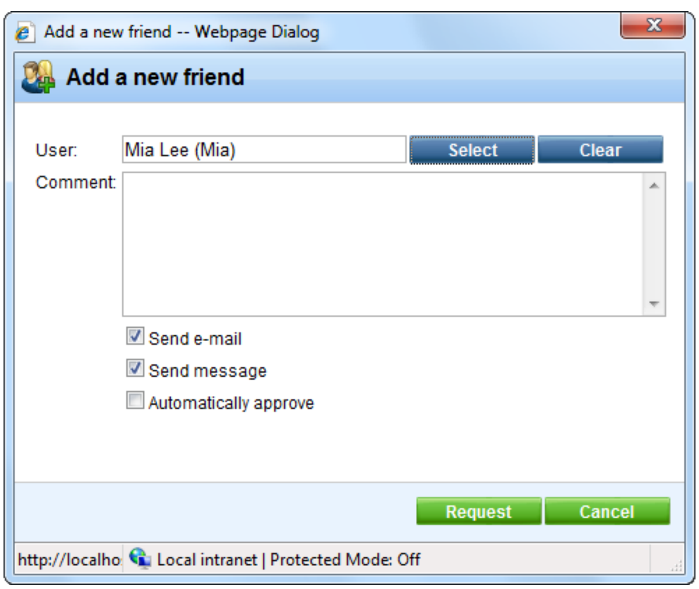


Add a new friend dialog

After clicking one of the options mentioned above, the **Add a new friend** dialog appears. After selecting a user (not possible in case of request from context menu) and filling in a comment explaining your request, there are two important options defining how the user will be notified about the request:

- **Send e-mail** - the user will be notified by a standard e-mail message
- **Send message** - the user will be notified by a Messaging module message
- **Automatically approve** - global administrators and users with the **Manage** permission for the Friends module can use this check-box, which creates the requested friendship without the other user's approval; not available on the live site

In both cases, the **Friend request** e-mail template mentioned in the section below is used as the body of the message.



Related e-mail templates

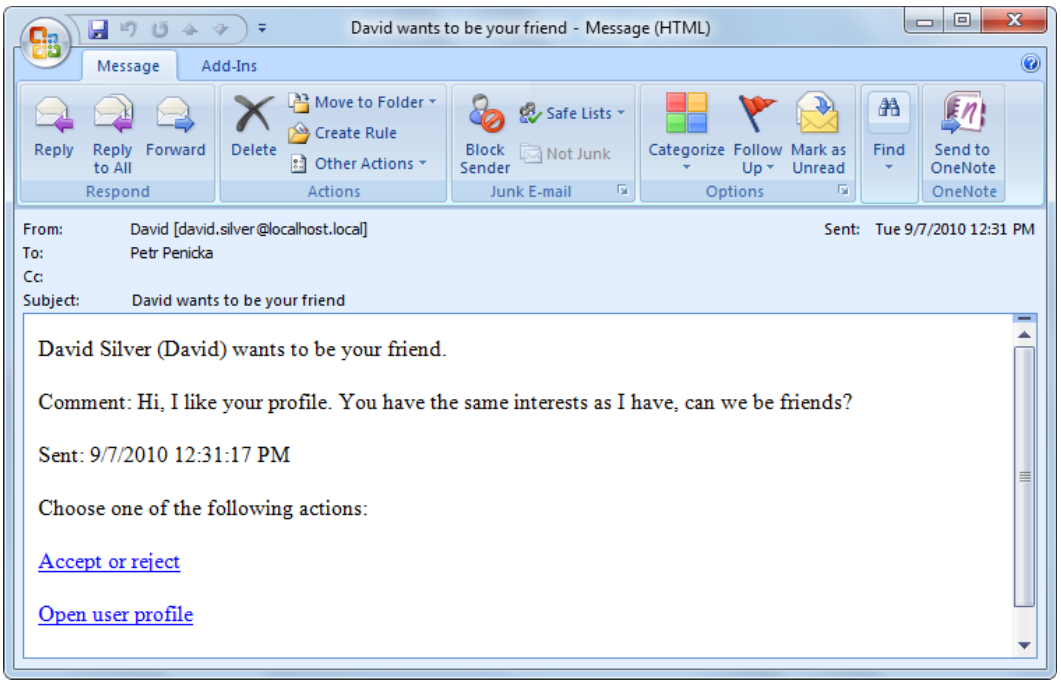
In **Site manager -> Development -> E-mail templates**, you can find the following e-mail templates related to the Friends module:

- **Friend approval** - template for e-mail or message confirming that a user approved your friendship request
- **Friend rejection** - template for e-mail or message confirming that a user rejected your friendship request
- **Friend request** - template for e-mail or message notifying you about the fact that some user requested your friendship

If you choose to **Edit** ( ) some of the templates, you will see two large text fields:

- **Text** - text of the template used for e-mails
- **Plain text** - text of the template used for Messaging module messages

Links in these e-mails need to be handled by a special page containing the **Friendship management** web part, which is a special web part that ensures handling of these requests.



8.21.3 Friends management

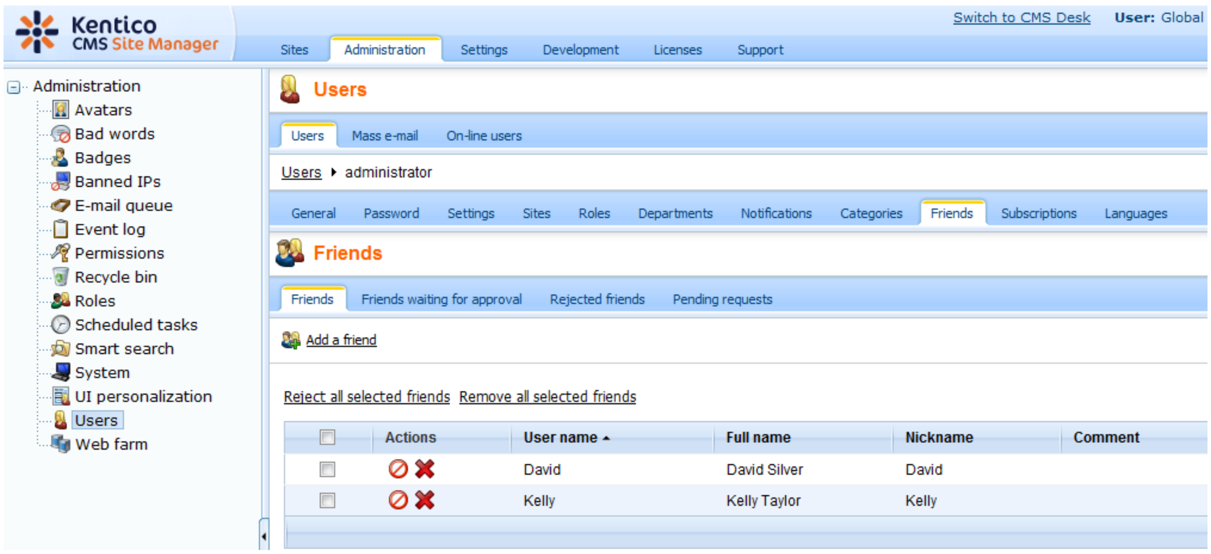
Users with access to **CMS Desk -> My desk -> My friends** can manage their own friendships from within this section of the administration interface.

Site administrators can manage all users' friends in both **CMS Desk** and **Site Manager**, in the **Administration -> Users** section, after choosing **Edit** ( ) next to some user and switching to their **Friends** tab.

In both cases, the four tabs described below are available. On the live site, the same functionality can be provided by the web parts described in the Available web parts topic.

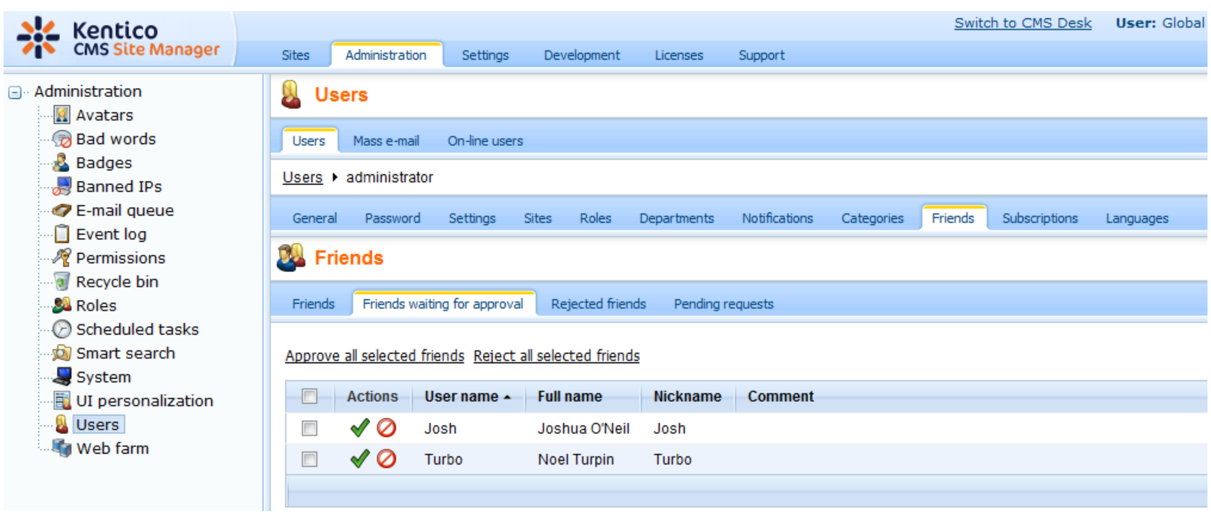
Friends (My friends)

On this tab, you can see a list of all your current friends. New friendships can be requested using the **Add a friend** link. You also can **Remove** () or **Reject** () friends in the list. The difference between the two is that rejected friends can't request your friendship anymore while they are in the rejected status, while removed friends can request the friendship again. By checking some of the check-boxes and clicking one of the **Reject all selected friends** and **Remove all selected friends** buttons, you can reject or remove more friends at once.



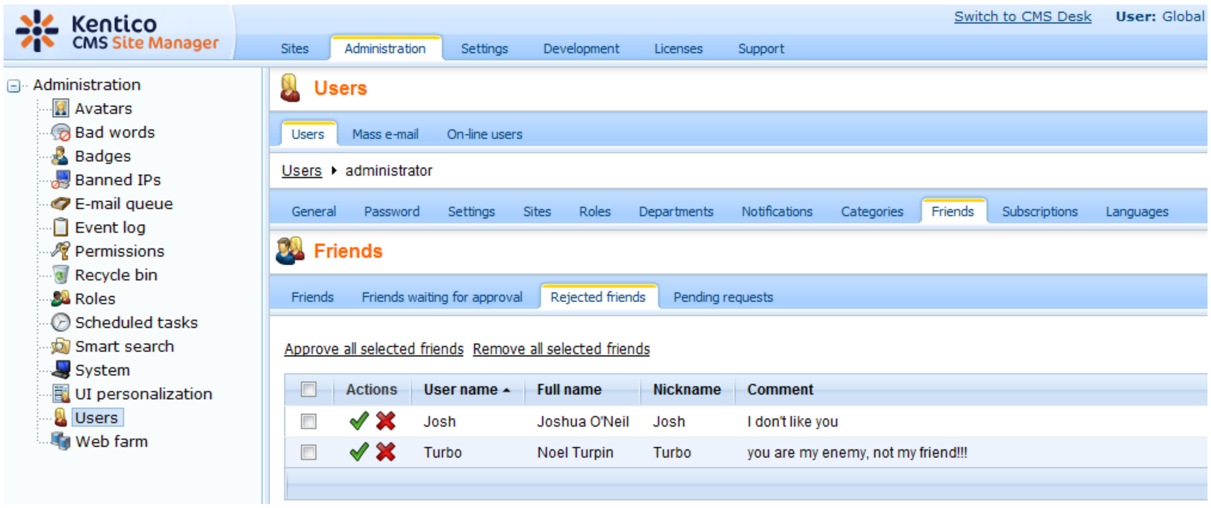
Friends waiting for approval (Friends waiting for my approval)

On this tab, you can see a list of all users who requested your friendship. You can either **Approve** () or **Reject** () their request. By checking some of the check-boxes and clicking one of the **Approve all selected friends** and **Reject all selected friends** links, you can approve or reject more requests at once.



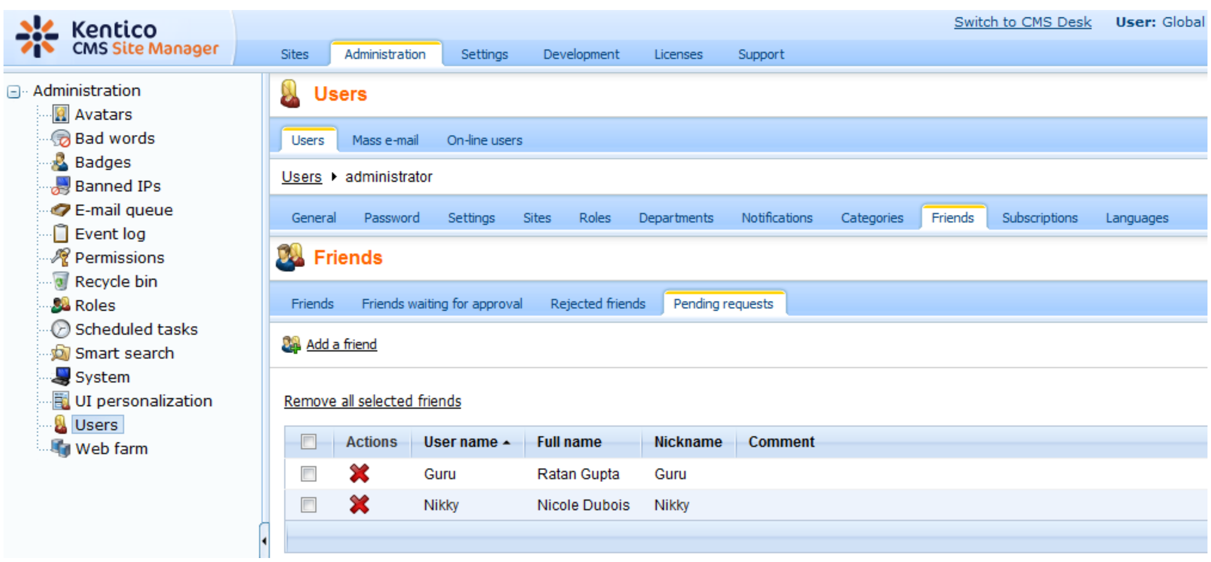
Rejected friends

On this tab, you can see a list of all users whose friendship you rejected. Once in the rejected status, the user can't request your friendship anymore. You can either **Approve** () some user's friendship, which makes them your friend, or **Remove** () the user from the list of rejected, which enables them to request your friendship again. By checking some of the check-boxes and clicking one of the **Approve all selected friends** or **Remove all selected friends**, you can approve or remove more users at once.



Pending requests (My pending requests)

On this tab, you can see a list of friends whose friendship you requested. New friendships can be requested using the **Add a friend** link.You can cancel a request by clicking the **Remove** () icon or you can select more users using the check-boxes and click the **Remove all selected friends** link, which cancels more friendship requests at once.

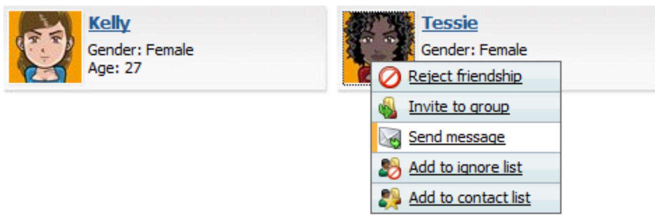


8.21.4 Available web parts

The Friends module comes with the following web parts. All of them are located in the **Community -> Friends** web part category. This page brings just a brief overview of them. For a detailed description of each web part's properties please see Kentico CMS Web parts reference.

Friends viewer

This web part can typically be used on user's profile page for viewing their friends. If you right-click the user's avatar image, you will be offered several tasks, as you can see in the screenshot below.



The following web parts are actually on-site equivalents of the friends management administration interface described in the [Friends management](#) topic:

Friends list

This web part displays a list of all friends of the current user.

Reject all selected friends Remove all selected friends

	Actions	User name ▲	Full name	Nickname	Comment	Approved
☐		Kelly	Kelly Taylor	Kelly		12/18/2008 6:42:46 PM
☐		Tessie	Iman Teshome	Tessie		11/21/2008 3:41:16 PM
					Items per page:	25

Friends waiting for approval

This web part displays a list of all users waiting for the current user's friendship approval.

Approve all selected friends Reject all selected friends

	Actions	User name ▲	Full name	Nickname	Comment	Requested
☐		Mia	Mia Lee	Mia		9/13/2010 12:44:49 PM
☐		Tessie	Iman Teshome	Tessie		9/13/2010 12:44:36 PM
					Items per page:	25

Rejected friends

This web part displays a list of all users whose friendship the current user rejected. These users can't request the current user's friendship while they are in this list.

Approve all selected friends Remove all selected friends

	Actions	User name ▲	Full name	Nickname	Comment	Rejected
☐		Josh	Joshua O'Neil	Josh	I don't like you	9/13/2010 12:30:40 PM
☐		Turbo	Noel Turpin	Turbo	you are my enemy, not my friend!!!	9/13/2010 12:31:26 PM
					Items per page:	25

My pending requests

This web part displays a list of all users whose friendship the current user requested.

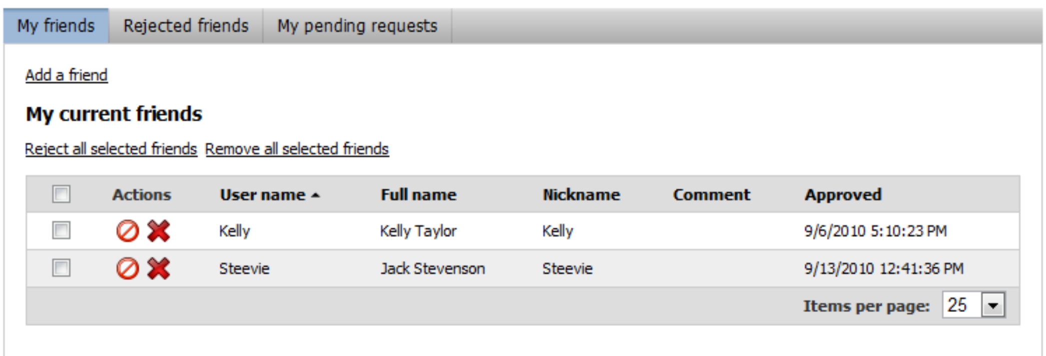
Request friendship

This web part displays only a link that, when clicked, opens the **Add a new friend** dialog.

Request friendship

My friends

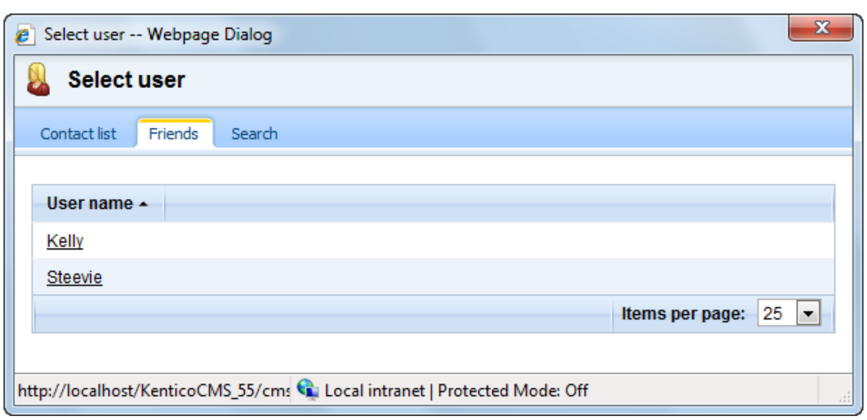
This web part combines the five previously mentioned web parts in one single web part. It's functionality is fully equal to the UI in **CMS Desk -> My Desk -> My friends**.



8.21.5 Examples of use

Messaging module

When selecting a recipient of a messaging module message, you can conveniently switch to the **Friends** tab and select one of your friends just by clicking their user name.



Fields visibility

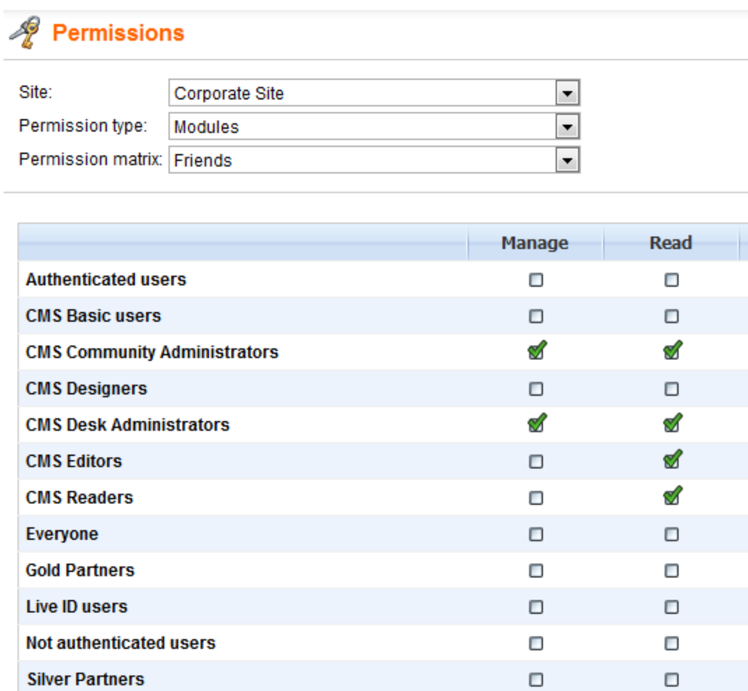
Users can set the visibility of some fields on their public profile to **Display to friends**, which makes the field visible only to their friends. For more information on this feature, please refer to [Development -> Membership, security and permissions -> Custom fields visibility].

8.21.6 Security

Permissions of the Friends module can be set in both **Site manager** and **CMS Desk**, in the **Administration -> Permissions** section. There, you need to select the **Modules -> Friends** permission matrix.

The following permissions can be assigned to particular roles:

- **Manage** - members of the role can manage Friends of particular users in *CMS Desk -> Administration -> Users*
- **Read** - members of the role can view friends of particular users in *CMS Desk -> Administration -> Users*

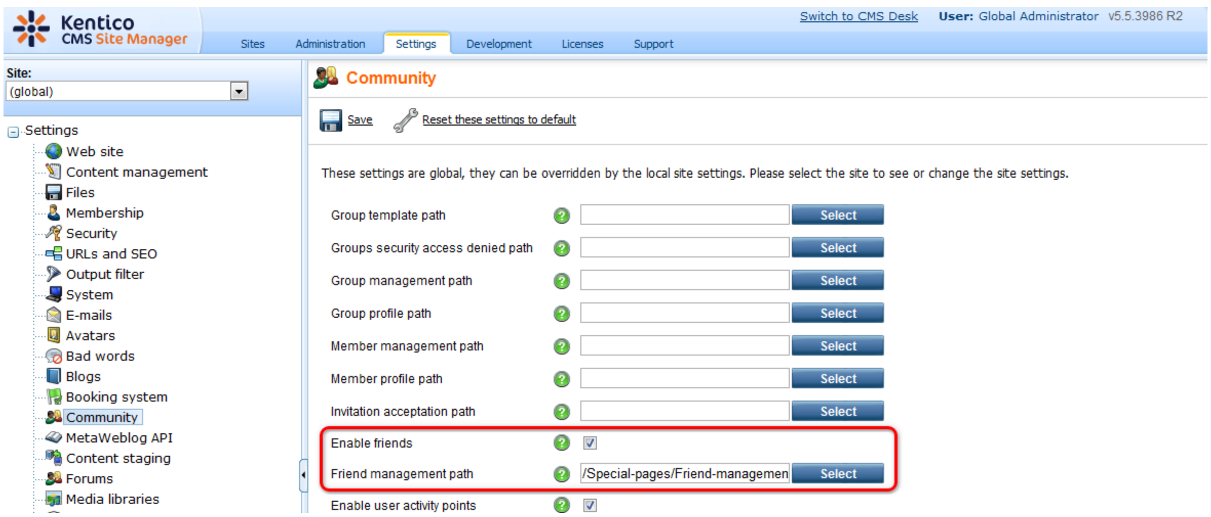


8.21.7 Settings

The following two settings available in **Site Manager -> Settings -> Community** are related to the Friends module:

Enable friends	Indicates if the Friends functionality should be available in other modules on the live site.

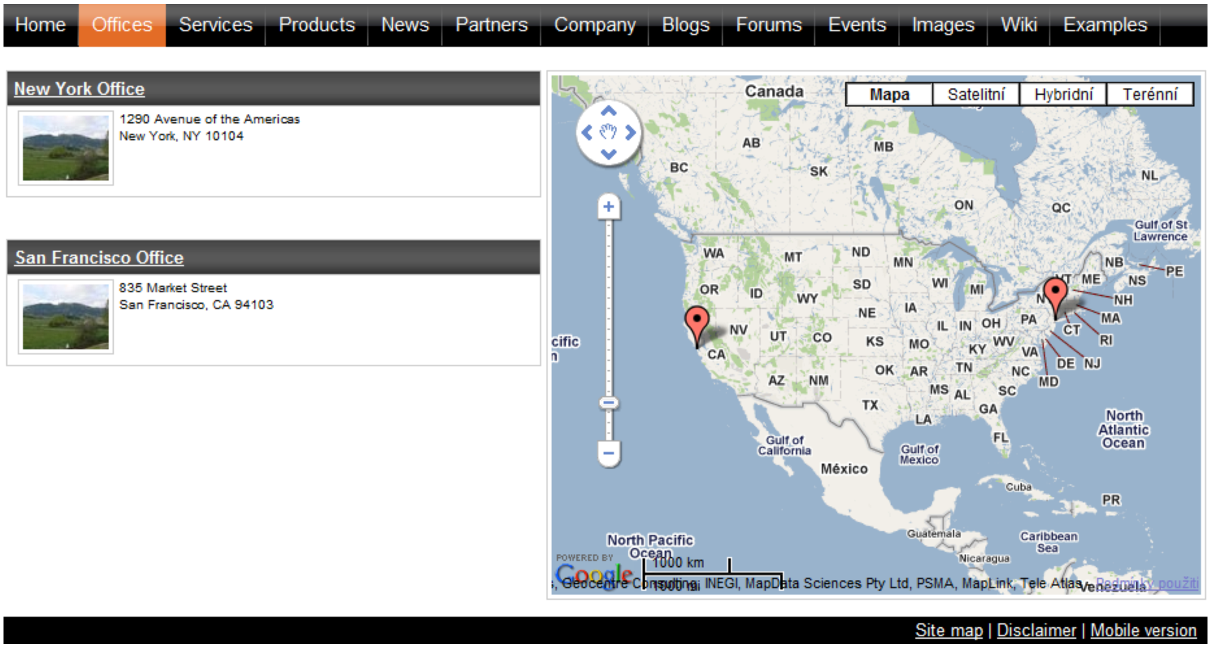
Friend management path	Node alias path of the page on that the **Friendship management** web part is placed. This is a special web part that handles friendship management actions from friendship request e-mails. If some user receives an e-mail with friendship request, there are links for friendship approval or rejection included in the mail. When the user clicks one of the links, they are redirected to this page and the user guid and type of action is transferred in form of querystring parameters. The **Friendship management** web part processes the received parameters and performs the necessary tasks for friendship approval or rejection. On the **Community Starter Site**, the *Special-pages/Friend-management* page serves exactly this purpose. More information on how such page can be created can be found in Community Site Guide -> Part 2 -> Creating the Special pages -> Creating the friend management page.



8.22 Geo mapping

8.22.1 Overview

The Geo mapping module allows you to display maps on your site and also display content on these maps. The module has no administration interface and consists only of 3 web parts and 3 widgets derived from these web parts. The web parts or widgets can be found under the **Maps** category in the web part or widget selection dialog.



You can see these web parts configured and working on the sample Corporate Site, under the **/ Examples/Webparts/Maps** node of the content tree.

What it is good for

The module can be used for many scenarios in which a geographical position is the key information:

- show your offices
- show your stores
- show your partners
- show real estates you offer
- etc.

You can use it to display virtually any content that has a location. The only requirement is that you tag your content with longitude and latitude.

An example of how to display content on the map can be found in this topic.

How to get longitude and latitude for a given place

You can use some on-line service that allows you to enter the country, city and street and shows you its longitude and latitude. One such service is available for example at http://world.maporama.com.

Available providers

You can choose from three different map providers:

- Bing Maps (originally Live Maps)
- Google Maps
- Yahoo Maps

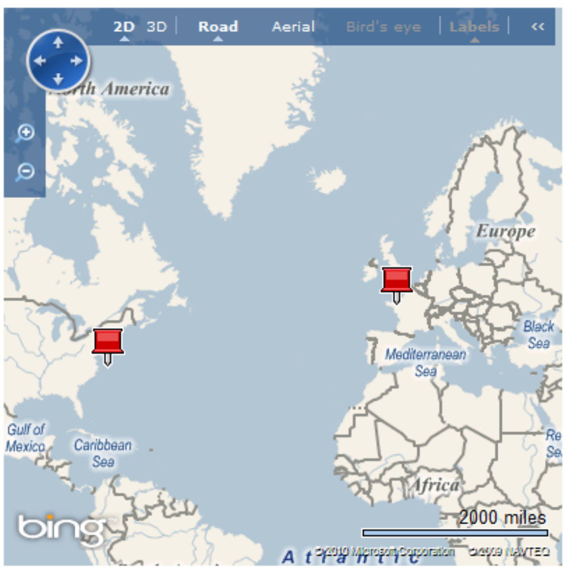
Each of these providers has its dedicated web part and widget. Click on the particular map provider to learn more.

8.22.2 Bing maps

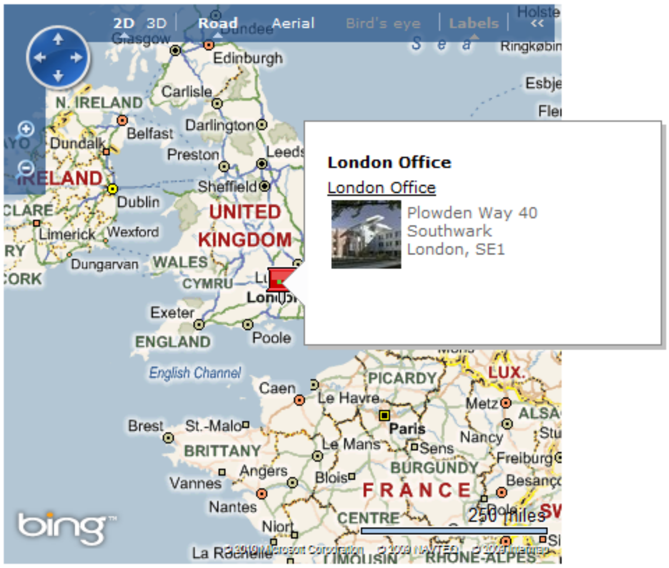
The **Bing maps** web part enables displaying of a map with defined location markers using the Bing maps service. It uses **Bing maps API**, which is described in detail in the following location: http://www.microsoft.com/maps/isdk/ajax/.

The **Bing maps** widget is derived from the web part. It provides the same functionality and look, while only a limited set of properties can be configured.

You can find an example of use of the web part on the Corporate Site sample website, on the **/Examples/WebParts/Maps/Bing maps** page. In the screenshot below, you can see how the web part looks on page load. It shows the location markers (learn here how to display them on the map). The map is zoomed according to the settings made by the **Large view scale** property and its middle point is on the coordinates specified by the **Default latitude** and **Default longitude** properties.



The screenshot below shows how the web part looks when one of the location markers is clicked. The map gets zoomed according to the value of the **Detailed view scale** property and a tooltip is displayed, showing the heading loaded from the document type field specified in the **Tooltip field** property and the rest of the content loaded using the transformation specified in the **Transformation** property.



If you enable the **Enable keyboard shortcuts** property, the following shortcuts can be used to control the map:

- R - switch to Road view
- A - switch to Aerial view
- H - switch to Aerial view with labels
- O - switch to Bird's eye view
- B - switch to Bird's eye view with labels

- Plus Sign (+) - zoom in
- Minus Sign (-) - zoom out

Detailed descriptions of all properties of the web part can be found in Kentico CMS Web Parts Reference or directly in the Web part properties dialog after clicking the **Documentation** link at the top right corner of the dialog.

8.22.3 Google maps

The **Google maps** web part enables displaying of a map with defined location markers using the Google maps service. It uses **Google maps API v3**, which is described in detail in the following location: http://code.google.com/intl/cs-CZ/apis/maps/documentation/v3/controls.html.

The **Google maps** widget is derived from the web part. It provides the same functionality and look, while only a limited set of properties can be configured.

You can find an example of use of this web part on the Corporate Site sample website, on the **/Examples/WebParts/Maps/Google maps** page. In the screenshot below, you can see how the web part looks on page load. It shows the location markers (learn here how to display them on the map). The map is zoomed according to the settings made by the **Large view scale** property and its middle point is on the coordinates specified by the **Default latitude** and **Default longitude** properties.

The screenshot below shows how the web part looks when one of the location markers is clicked. The map gets zoomed according to the value of the **Detailed view scale** property and a tooltip is displayed, showing the heading loaded from the document type field specified in the **Tooltip field** property and the rest of the content loaded using the transformation specified in the **Transformation** property.

If you enable the **Enable keyboard shortcuts** property, the following shortcuts can be used to control the map:

- **Up Arrow** - move a little north
- **Down Arrow** - move a little south
- **Left Arrow** - move a little west
- **Right Arrow** - move a little east

- **Page Up** - move a large step north
- **Page Down** - move a large step south
- **Home** - move a large step west

- **End** - move a large step east

- **Plus Sign (+)** - zoom in
- **Minus Sign (-)** - zoom out

Detailed descriptions of all properties of the web part can be found in Kentico CMS Web Parts Reference or directly in the Web part properties dialog after clicking the **Documentation** link at the top right corner of the dialog.

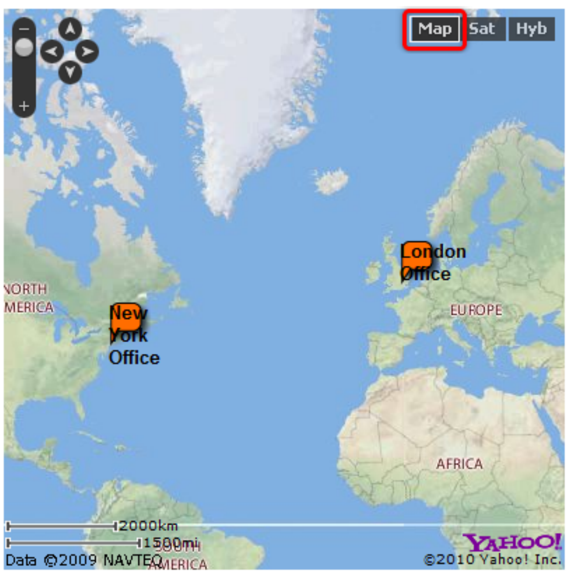
8.22.4 Yahoo maps

The **Yahoo maps** web part enables displaying of a map with defined location markers using the Yahoo maps service. It uses **Yahoo maps API**, which is described in detail in the following location: http://developer.yahoo.com/maps/ajax/

The **Yahoo maps** widget is derived from the web part. It provides the same functionality and look, while only a limited set of properties can be configured.

For Yahoo maps to work, you need to have a **Yahoo AppID** for your application and enter it into the **Yahoo AppID** property of the web part. You can obtain the AppID at http://developer.yahoo.com/wsregapp/. Yahoo account is needed in order to obtain the AppID. However, if you don't have one, you will be offered to create it after going to the link above.

You can find an example of use of this web part on the Corporate Site sample website, on the **/Examples/WebParts/Maps/Yahoo maps** page. In the screenshot below, you can see how the web part looks on page load after clicking the **Map** button. It shows the location markers (learn here how to display them on the map). The map is zoomed according to the settings made by the **Large view scale** property and its middle point is on the coordinates specified by the **Default latitude** and **Default longitude** properties.



The screenshot below shows how the web part looks when one of the location markers is clicked. The map gets zoomed according to the value of the **Detailed view scale** property and a tooltip is displayed, showing the heading loaded from the document type field specified in the **Tooltip field** property and the rest of the content loaded using the transformation specified in the **Transformation** property.

If you enable the **Enable keyboard shortcuts** property, the following shortcuts can be used to control the map:

- **Left Arrow** - pan the map west
- **Right Arrow** - pan the map east
- **Up Arrow** - pan the map north
- **Down Arrow** - pan the map south

- **Page Up** - zoom the map in
- **Page Down** - zoom the map out

Detailed descriptions of all properties of the web part can be found in Kentico CMS Web Parts Reference or directly in the Web part properties dialog after clicking the **Documentation** link at the top right corner of the dialog.

8.22.5 Example: Displaying offices on the map

This example will show you how to display a list of offices and their location on the map. We will use the **Google maps** web part in this example. However, the procedure is identical for **Bing maps** and **Yahoo maps**, too.

We will use the Corporate Site sample website for this example, and create a page for offices directly under the website root. The page will follow the same concept as the **Company/Offices** page on this site. We will create the new one only so that we can see the step-by-step procedure of creating such a page from scratch.

How to do it in general

To display content on a map, you need to have this content stored in the content tree as standard documents. These documents can be of any document type as long as the **document type contains two fields for the geographical coordinates** - one for **longitude** and one for **latitude**.

Then you need to configure the properties of the maps web part to display documents of this document type stored within the desired location, and specify which fields of the document type contain the coordinates. This ensures displaying of the location markers. Finally, you need to choose a field for the heading and a transformation for the content of the tooltip displayed when a location marker is clicked.

Step 1 - Creating a new page with offices

1. Sign in as an administrator to **CMS Desk -> Content** and click the root of the content tree.

2. Click **New** and choose to create a new **Page (menu item)**. Enter *Offices* for **Page name** and choose to **Create a blank page** using the **Two columns** layout. Click **Save**.

3. First we need to add a list of offices. Switch to the **Design** tab and add the **Listings and viewers/ Repeater** web part to the **zoneLeft** web part zone. Set the following properties:

- **Path**: /{0}/%
- **Document types**: CMS.Office
- **Transformation**: CMS.Office.Simple
- **Selected item transformation**: CMS.Office.Simple
- **Item separator**:
- **Content before**: <div class="GeneralList">
- **Content after**: </div>

Click **OK**.

Step 2 - Geo-coding your information

4. Now we will create two documents of the **Office** type. Click the **Offices** page and click **New**. Choose to create a new **Office** and enter the following values:

- **Office name**: New York Office
- **Address line 1**: 1290 Avenue of the Americas
- **City**: New York
- **ZIP code**: 10104
- **State**: NY
- **Country**: USA
- **Phone**: 123456789
- **E-mail**: ny@example.com
- **Latitude**: 40.76
- **Longitude**: -73.98
- **Office photo**: choose some image from your local disk

Please note

The Office document type already contains the **Latitude** and **Longitude** fields. These are the fields that you will specify later in the map web part properties as the **source for the geographical position** of the location markers. If you're using a custom document type, you will need to define these fields manually. They must be of the **Decimal number** type. You can call them as you need, you only need to specify the appropriate name in the web part properties.

Values are valid in the following intervals:

- **Latitude**: from -90 to 90
- **Longitude**: from -180 to 180

Click **Save**.

5. Create another office:

- **Office name**: San Francisco Office
- **Address line 1**: 835 Market Street
- **City**: San Francisco
- **ZIP code**: 94103
- **State**: CA
- **Country**: USA
- **Phone**: 123456789
- **E-mail**: sf@example.com
- **Latitude**: 37.78
- **Longitude**: -122.41
- **Office photo**: choose some image from your local disk

Click **Save**.

Step 3 - Displaying the content on the map

When you see the page now, it displays only a list of offices. Switch to the **Design** tab and add the **Maps/Google maps** web part to the **zoneRight** web part zone.

First, we will configure which documents should be displayed on the map. Set the following properties:

- **Path:** /{0}/%
- **Document types:** CMS.Office

It ensures that all offices in the current site section will be shown. Now we specify the transformation used for the text displayed in the balloon:

- **Transformation:** CMS.Office.Preview

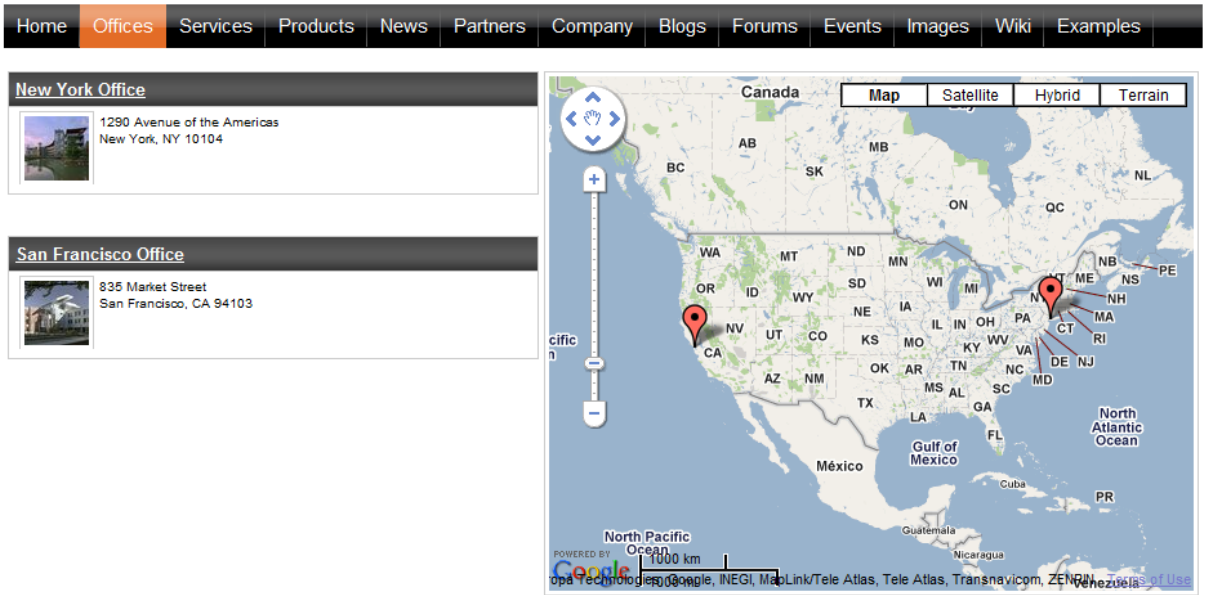
Now we set the following values that specify how the map is displayed:

- **Large view scale:** 3 (the zoom used on page load)
- **Detailed view scale:** 10 (the zoom used a location marker is clicked)
- **Width:** 500 (in pixels)
- **Height:** 400 (in pixels)
- **Default latitude:** 39.27 (latitude of the map center when the overview map is displayed on page load)
- **Default longitude:** -98.20 (longitude of the map center when the overview map is displayed on page load)
- **Latitude field:** OfficeLatitude (the field of the document containing the latitude position)
- **Longitude field:** OfficeLongitude (the field of the document containing the longitude position)

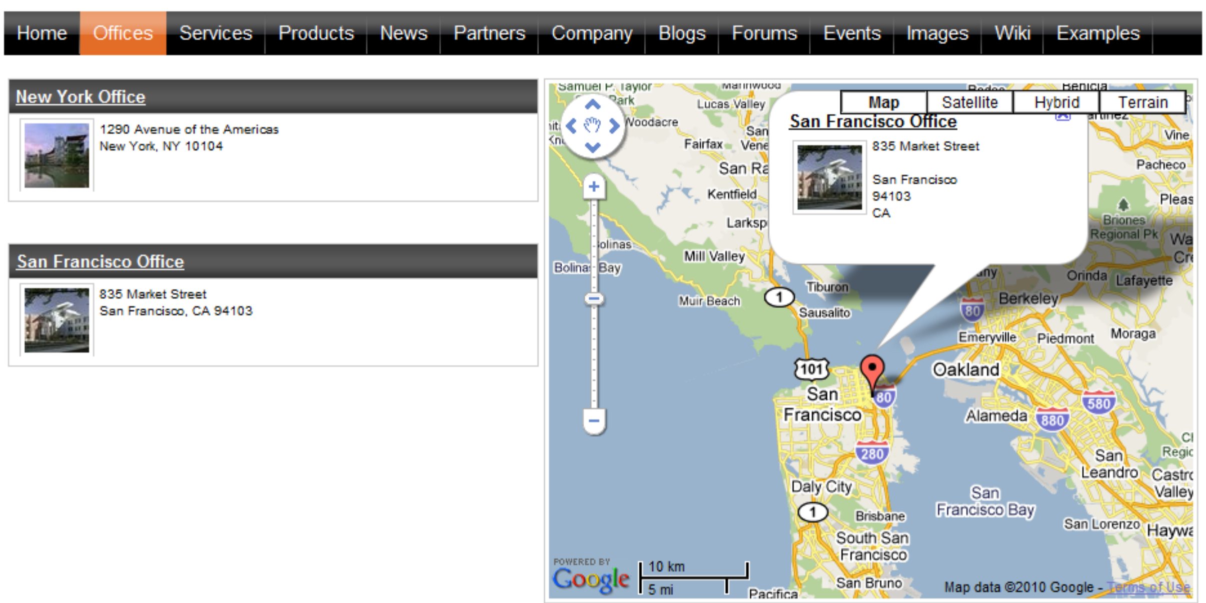
- **Tooltip field:** OfficeName (the field used for the heading of the tooltips)
- **Content before**: <div class="Map">
- **Content after**: </div>

Click **OK**.

Sign out and see the page. It will look like this:



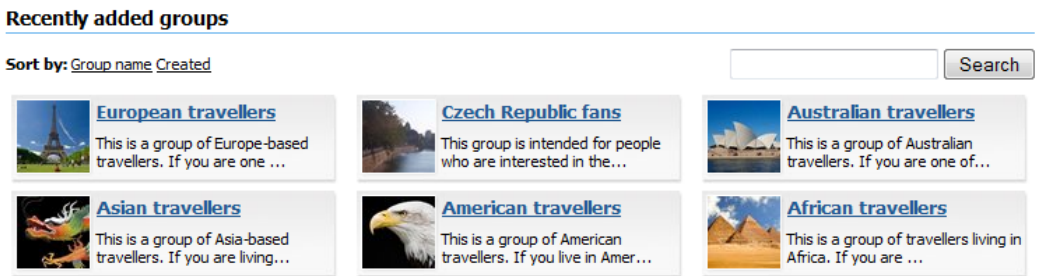
You can see two baloons on the map - these are the location markers. When you mouse-over a balloon, you will see the office name. When you click a balloon or an office link in the list, you will see the detailed view:



8.23 Groups

8.23.1 Overview

The groups module allows site members who share an interest in a certain topic or field to access related information and share their own experiences on a subsection of your website. Site users can create new groups or join already existing ones. Groups may contain their own documents section, forums, message boards, media libraries, polls and projects and have the option of defining group roles. Groups can also be useful for managing access control.



Site administrators can manage groups of a given site through the CMS Desk interface. Learn more in the Groups management topic.

Both site administrators and users who are in authorized roles can edit the content and various settings of groups. Learn more about this in the Editing a group topic. Further settings available only to global administrators are described in the Settings topic.

To allow users to interact with the groups module, you have to place some group web parts, which can be found in the **Community** web part category, somewhere on your site. An example of this that describes how to enable users to create groups can be found in the Enabling users to create groups topic. A full list of group web parts and their function and properties can be found in the Web Parts reference.

The Groups internals and API sub-chapter provides information about the database tables and classes used by the module and examples of how groups can be managed using the API.

Kentico CMS Community Site Guide contains some additional group examples and tutorials:

- Part 1 -> Groups: Examples of the functionality and customization of groups.
- Part 2 -> Creating the Groups section: A step-by-step tutorial on how to create a sample group pages section of a website.

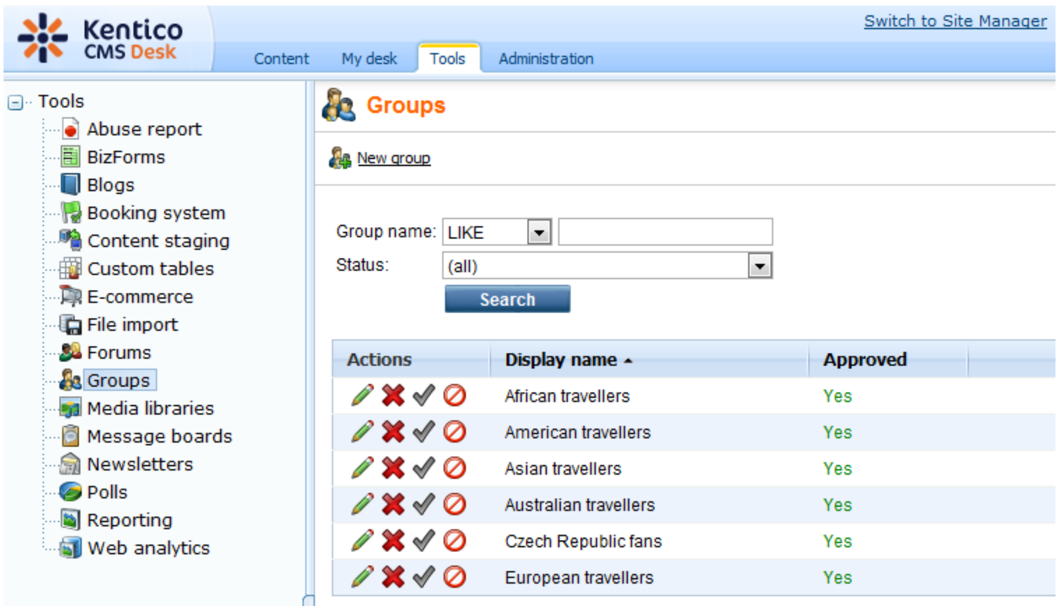
You will need to have the sample Community Site installed to follow Kentico CMS Community Site Guide.

8.23.2 Groups management

In the screenshot below, you can see the groups management interface located in **CMS Desk -> Tools -> Groups**. On this page, you can see a list of all groups on the site. You can filter displayed groups using the filter above the list. Filtration is possible by the group's **display name** and approval **Status**.

Even though groups are typically created by site users on the live site, you can create new groups in this section of the administration interface too. It can be done by clicking the **New group** link at the top part of the page. Groups in the list can be **Edited** () or **Deleted** (). The Editing a group topic describes editing in more detail.

If an administrator's approval is needed after a user creates a new group, the approval can be done by clicking the **Approve** () icon. By clicking the **Reject** () icon, groups can be switched back to the state they were in before they were approved. If you do this to an existing group, the group will not be displayed on the live site.



8.23.3 Editing a group

There are two ways how group properties can be edited:

- On-site management using the **Group profile** web part; this is typically used by group administrators
- Administration interface in **CMS Desk -> Tools -> Groups**, after choosing to **Edit** () a group; this is typically used by site administrators

Both of the two approaches provide the same tabs layout, with the difference that there are some extra settings in the administration interface compared to on-site management. These will be marked as **AI only** in the descriptions below.

General tab

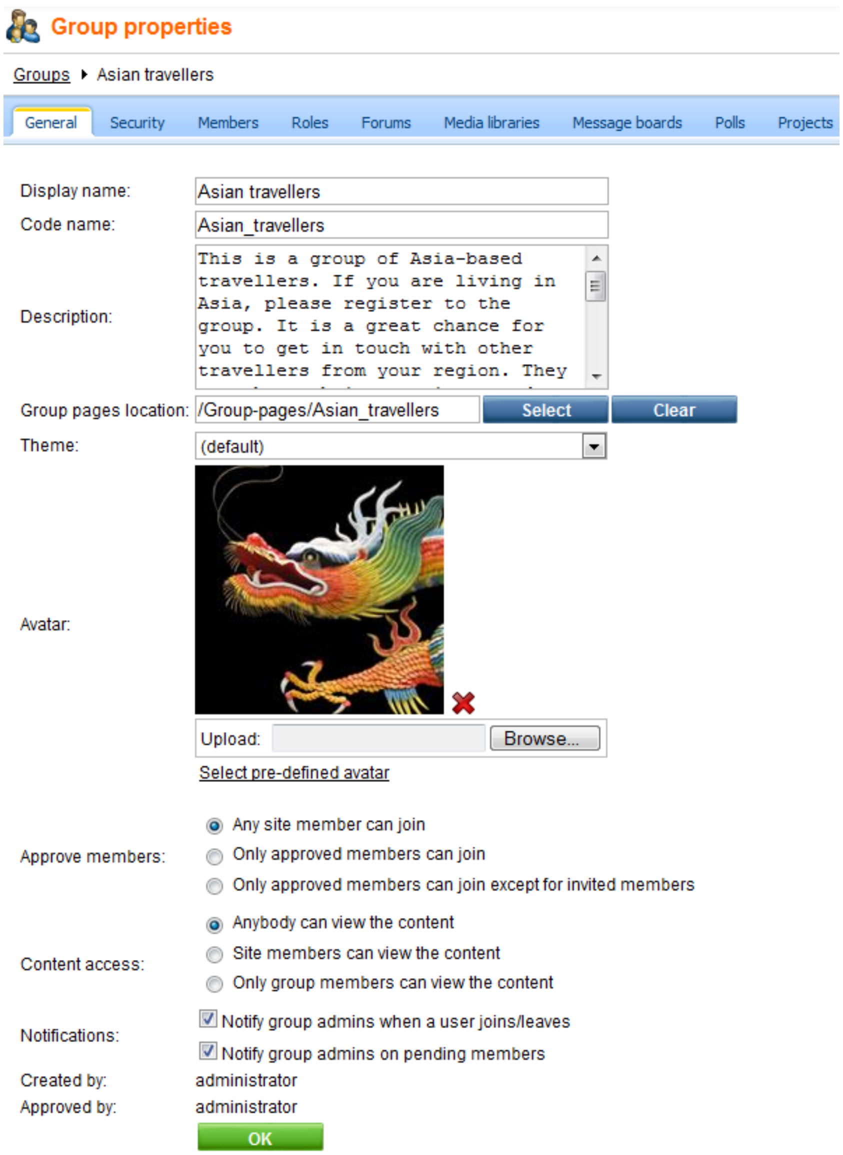
On this tab, you can set the general properties of the group:

- **Display name** - name of the group displayed on the site and in the administration interface; *AI only*
- **Code name** - name of the group used in site code; *AI only*
- **Description** - text describing the group
- **Group pages location** - node alias path of the location where group pages of the group are stored
- **Theme** - allows the selection of one of the website CSS stylesheets that will be used by the pages of the group

- **Avatar** - group avatar image

- **Approve members** - determines if users can join the group with or without a group admin's approval; the last option allows invited members to join without approval
- **Content access** - determines who can view content of the group pages
- **Notify group admins when a user joins/leaves** - if checked, group administrators will receive a notification e-mail when a user joins/leaves the group
- **Notify group admins on pending members** - if checked, group administrators will receive a notification e-mail when a user requests to join the group and admin's approval is needed

- **Created by** - displays who created the group
- **Approved by** - displays who approved the group to be created on the site



Security tab

On this tab, you can use the matrix to set permissions for group pages. The following permissions can be assigned:

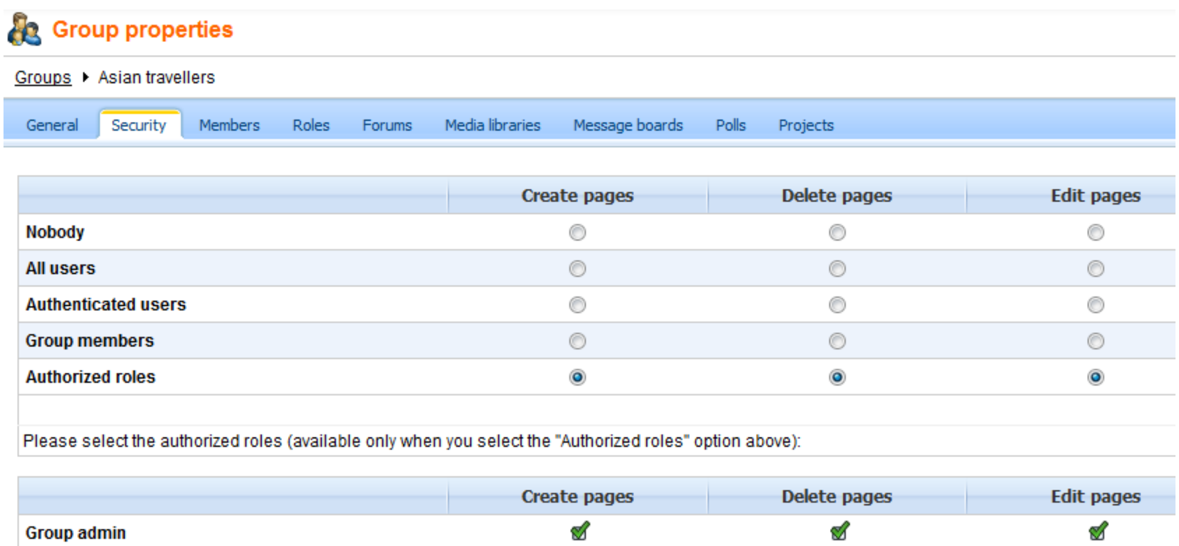
- **Create pages** - users can create group pages
- **Delete pages** - users can delete group pages
- **Edit pages** - users can edit group pages

These permissions can be assigned to:

- **Nobody** - nobody can perform the action
- **All users** - all users can perform the action
- **Authenticated users** - only signed-in users can perform the action, i.e. anonymous public users cannot perform it
- **Group members** - only group members can perform the action, i.e. authenticated non-group members and anonymous users cannot perform it
- **Authorized roles** - only members of the group roles selected below can perform the action

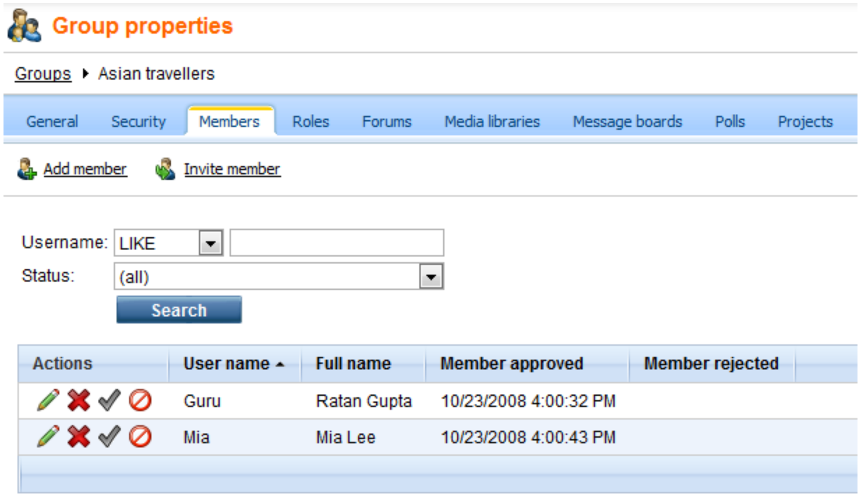
Group admin's permissions

Group administrators can perform any of these actions, even if they don't have the permissions assigned.



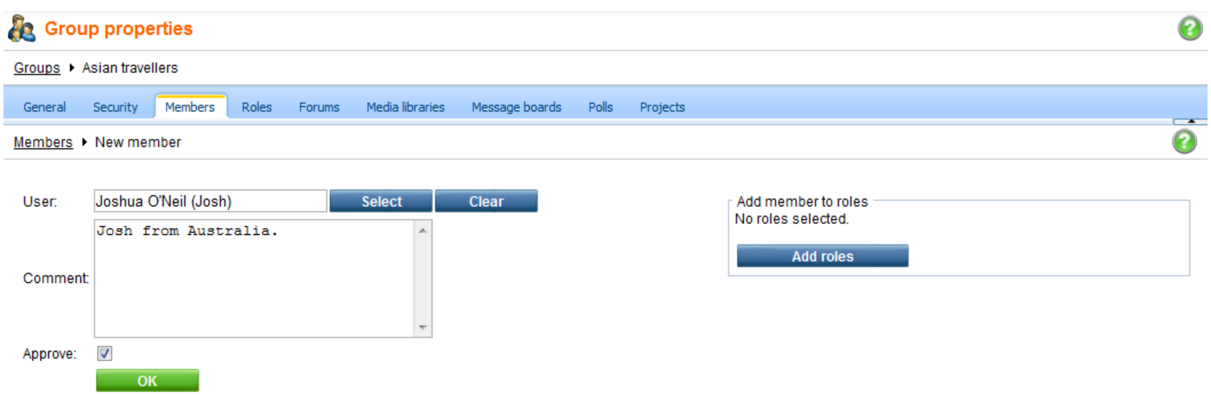
Members tab

On this tab, you can see a list of all members of the group. You can **Edit** ( ) or **Delete** ( ) users in the list. You can also **Approve** ( ) members' requests for joining the group or **Reject** ( ) them from the group. Once rejected, the user cannot request to join the group until they are approved again.



Clicking the **Add member** link, you can add users to the group directly, without sending an invitation to them. This is possible only in the administration interface. On the live site, only the **Invite member** link is displayed. When adding a user to a group, you have the following options:

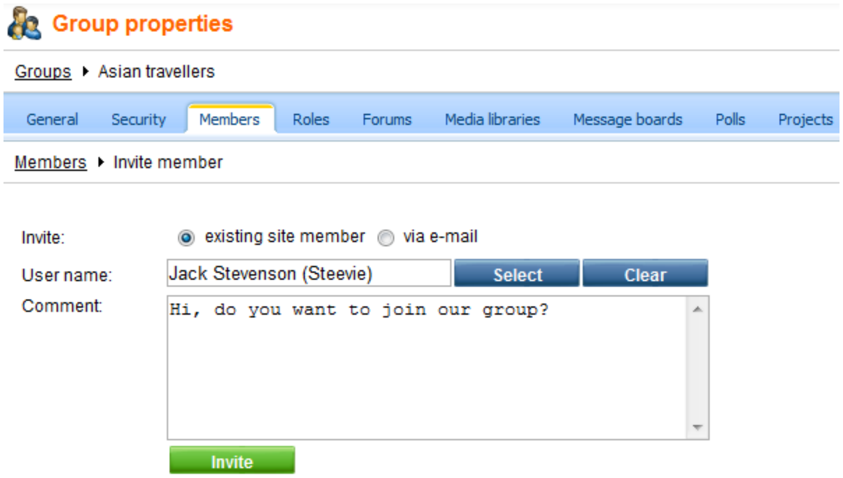
- **User** - select an existing site user who you want to add to the group
- **Comment** - text comment that you can add to the user; this comment is not sent to the user, it is only displayed in the administration interface
- **Approve** - if checked, the user will be automatically approved; if not, a user will need a group admin's approval before they become members of the group
- **Add roles** - you can use this button to assign the user to group roles; a dialog where you can select from a list of available group roles is opened



After clicking the **Invite member** link, the dialog displayed in the screenshot below will be displayed. There are two ways of invitation:

- **Invite existing site member** - after selecting an existing site user in the **User name** field, an e-mail will be sent to the user's e-mail address based on the **Groups - member invitation** e-mail template; text entered to the **Comment** field will be included in the e-mail; the user can join the group either by clicking a link in the e-mail, or via the **My sent invitations** web part

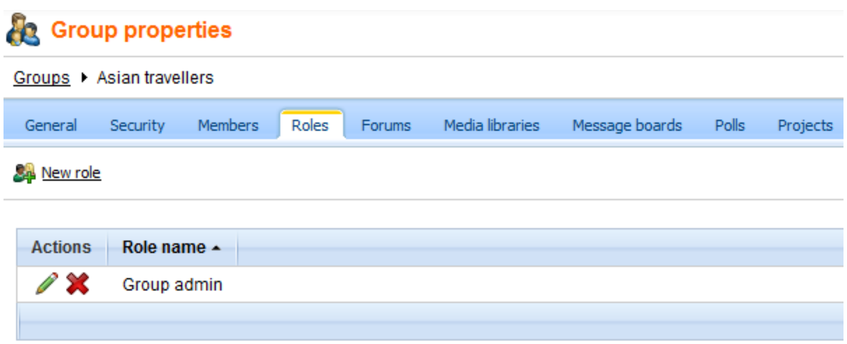
- **Invite via e-mail** - this way, you can send the invitation to any e-mail address that you enter into the **E-mail** field; in this case, user will be required to register to the site after clicking the join link in the e-mail; text entered to the **Comment** field will be included in the e-mail



Roles tab

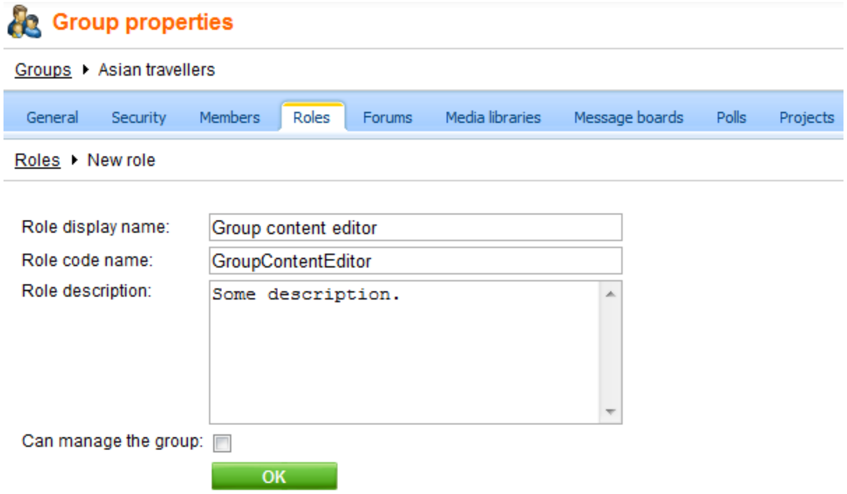
On this tab, you can see a list of roles defined for the group. These roles are applicable only in the context of the group. Don't confuse them with website roles, which can be set in **Site Manager -> Administration -> Roles**.

Roles in the list can be **Edited** ( ) and **Deleted** ( ).

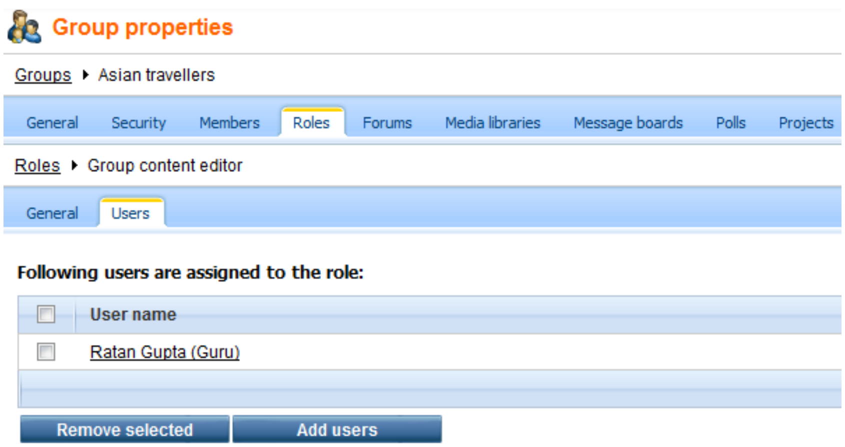


If you click the **New role** link above the list, you can define a new role for the group. The following properties can be entered:

- **Role display name** - name of the role that will be used on the live site and in the administration interface
- **Role code name** - name of the role used in your code; *AI only*
- **Role description** - text description of the role
- **Can manage the group** - indicates if members of the role can manage the group by means of the **Group profile** web part



When **Editing** (✎) a role, two tabs are offered. On the **General** tab, you can change the details entered when creating the role, as described above. On the **Users** tab, you can see a list of all members assigned to the role. These can be removed from the role by selecting the checkbox to the left of users in the list and clicking the **Remove selected** button. New users can be added to the role using the **Add users** button.



Forums tab

On this tab, you can create and manage the group's forums. As these forums are standard Kentico CMS forums set into the context of the group, please refer to the Modules -> Forums chapter of this guide for more information on their management.

Media libraries tab

On this tab, you can create and manage the group's media libraries. As these are standard Kentico CMS media libraries set into the context of the group, please refer to the Modules -> Media libraries chapter of this guide for more information on their management.

Message boards tab

On this tab, you can manage the group's message boards. As these are standard Kentico CMS message boards set into the context of the group, please refer to the Modules -> Message boards chapter of this guide for more information on their management.

Polls tab

On this tab, you can manage the group's polls. As these are standard Kentico CMS polls set into the context of the group, please refer to the Modules -> Polls chapter of this guide for more information on their management.

Projects tab

On this tab, you can manage the group's projects. As these are standard Kentico CMS projects set into the context of the group, please refer to the Modules -> Project management chapter of this guide for more information on their management.

8.23.4 Enabling users to create groups

You can enable site users to create new groups by placing the **Community -> Group registration** web part on your site. You have to set the following properties of the web part:

- **Template source alias path** - alias path of the document that will be used, together with the documents stored under it, as a template for groups created by the web part; if left empty, the value of the *Site Manager -> Settings -> Community -> Group template path* field will be used here
- **Template target alias path** - alias path where the documents copied from the *Template source alias path* will be loaded when a group is created
- **Automatically create forum** - if checked, a forum group and a General discussion forum are automatically added under the created group
- **Automatically create media library** - if checked, a media library is automatically added under the created group
- **Automatically create smart search indexes** - if checked, a smart search index is automatically created for the documents of the created group, as well as for the new forum if the *Automatically create forum* property is enabled
- **Group profile URL path** - alias path of the page containing the group profile; the *{groupname}* wildcard can be used to substitute for the name of the current group
- **Combine with default culture** - if checked, default culture will be used when creating group pages under a culture where the source or target nodes were not found
- **Group name label text** - text that will be displayed in the form before the field where the group name is entered
- **Text after successful registration** - text displayed when a group is successfully created
- **Text after successful registration with approving** - text displayed when a group is successfully created, but requires an administrator's approval to be published on the web

- **Require approval** - if checked, the group will have to be approved by a site administrator before it is published on the site
- **Redirect to URL** - URL where the user will be redirected after creating the group
- **Hide form after registration** - if checked, the form will be hidden after creating the group

Group pages templates

Each group has its own section on the website where its content is stored - so called group pages.

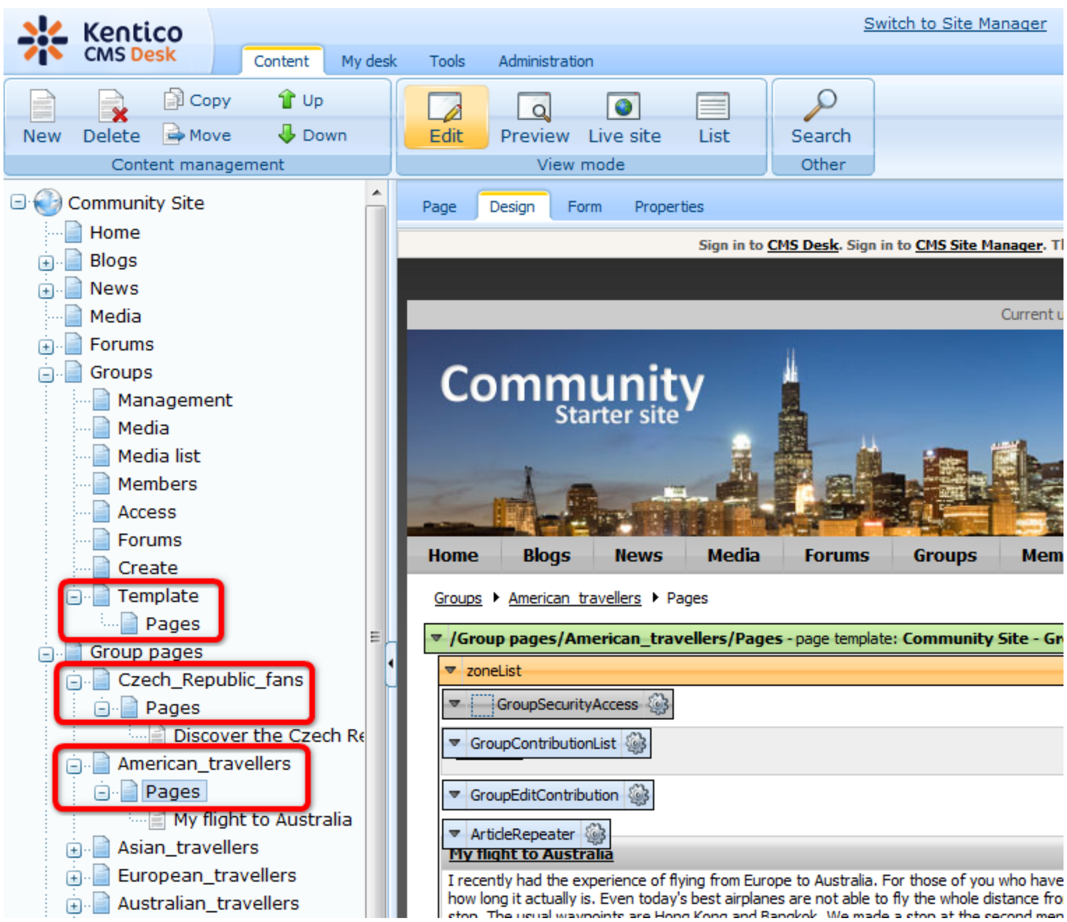
When adding the **Group registration** web part to your site, you have to specify the **Template source alias path** and **Template target alias path** properties. These two properties are essential when creating the **group pages section** of each group.

The page specified by the **Template source alias path** and all its sub-pages are copied to the location specified by the **Template target alias path**.

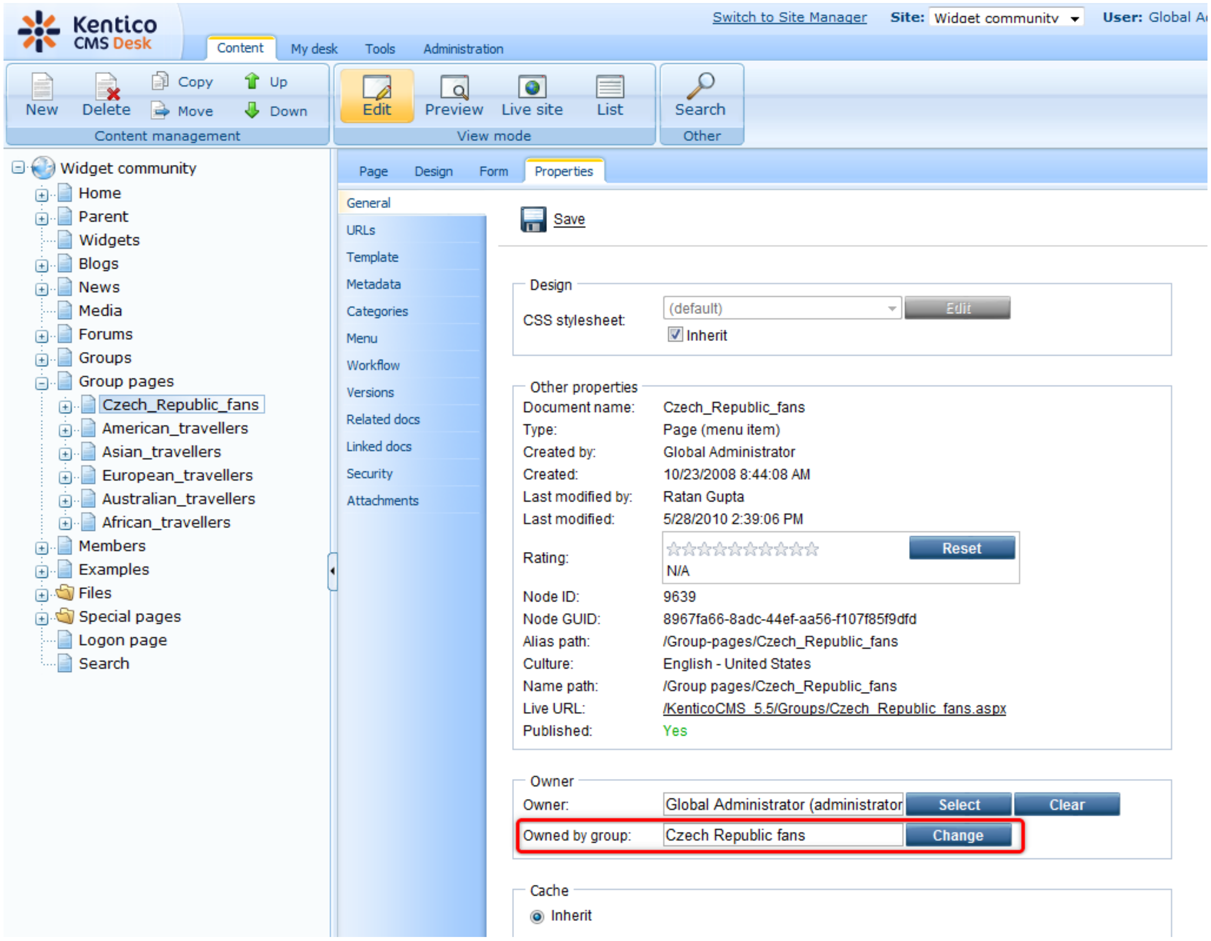
To get a better idea of how this works, you can take a look at our sample **Community Starter site**. On the site, the **Group registration** web part is configured the following way:

- **Template source alias path**: /Groups/Template
- **Template target alias path**: /Group-pages

As you can see in the screenshot below, there is the **/Groups/Template** page with one sub-page: **Pages**. When a new group is created, its title page is created under **/Group-pages** and the **Pages** page is created under it. As you have probably noticed, the web parts placed on the title page are identical to those placed on the **Template** page. Web parts on the **Pages** page are also identical to the source **Pages** page. Under **Pages**, all group documents will be stored.



All Kentico CMS documents can be set to be owned by a group. This can be done by signing into CMS Desk, selecting a document from the content tree, switching to **Edit** mode and going to **Properties -> General -> Owned by group** and choosing a group from the drop-down list displayed after clicking the **Change** button. This property is used to identify which group a document belongs under and is used by various group context sensitive web parts to display the correct content. It also influences the editing permissions of Group administrator widget zones.



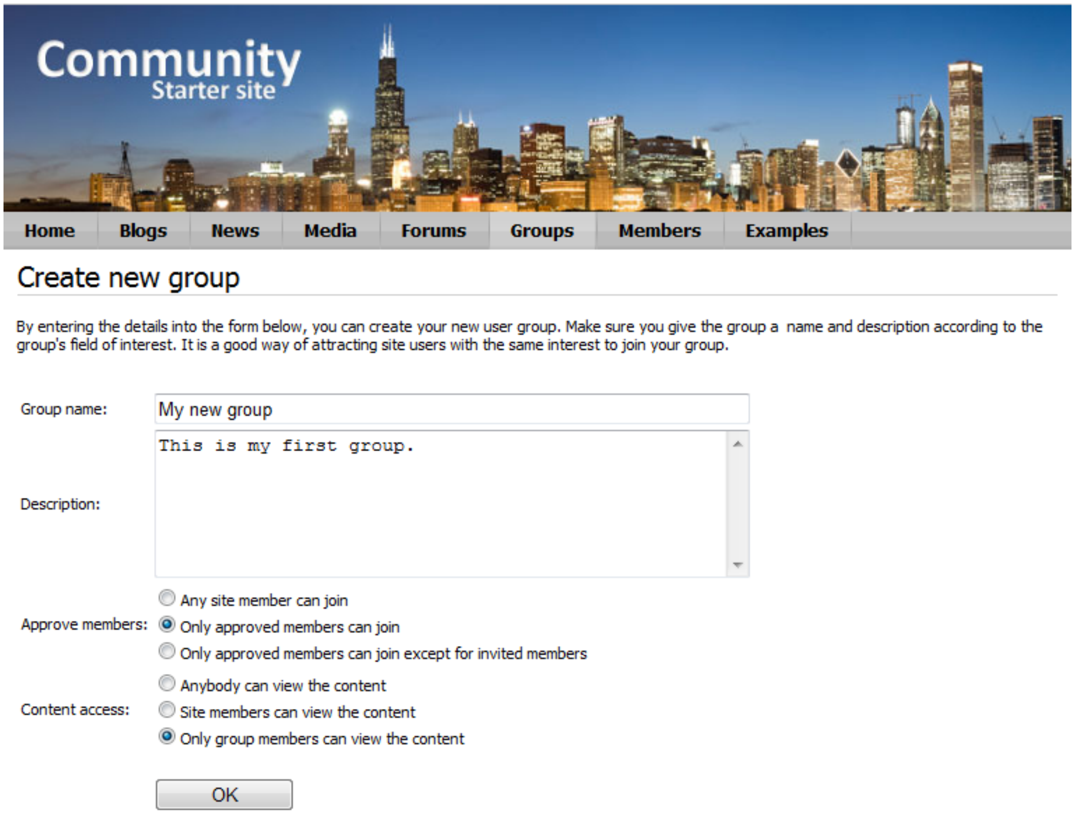
Please refer to Community Site Guide -> Part 2 -> Creating the Groups section for a step-by-step tutorial on how to create a sample group pages section of a website.

8.23.5 How site users create a new group

When a user wants to create a new group on the live site using the **Group registration** web part, they have to fill in the following details:

- **Group name** - name of the group displayed on the site and in the administration interface
- **Description** - text describing the group
- **Approve members** - determines if users can join the group with or without group admin's approval; the last options allows invited members to join without the approval
- **Content access** - determines who can view content of the group pages

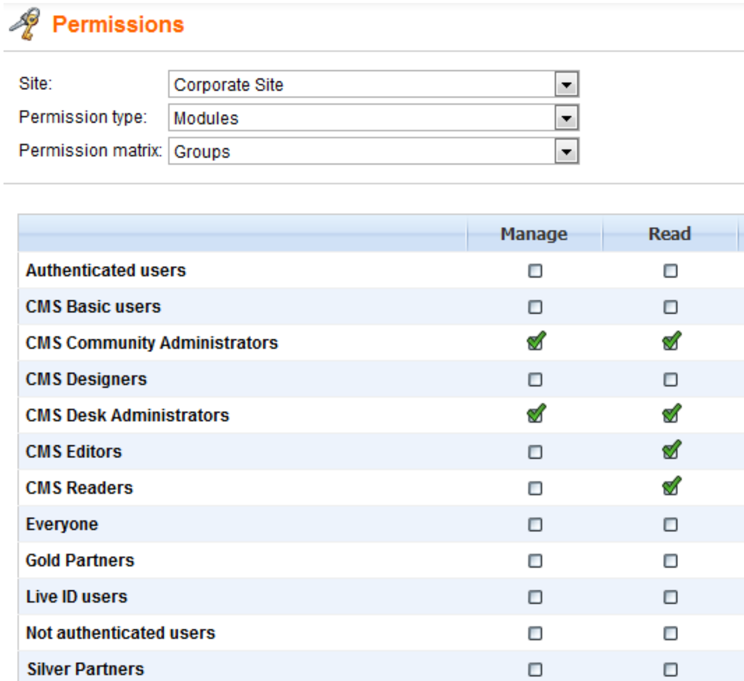
After clicking **OK**, the group will be created and group pages added to the site. In case that site administrator's approval is needed, these actions will be performed after the approval.



8.23.6 Security

Permissions of the Groups module can be set in **Site Manager -> Administration -> Permissions**. The following permissions can be assigned to members of the particular roles:

- **Manage** - allows managing of groups via the administration interface
- **Read** - allows viewing group settings, but does not allow any changes to be made to them



Permissions

Site: Corporate Site
Permission type: Modules
Permission matrix: Groups

	Manage	Read
Authenticated users	☐	☐
CMS Basic users	☐	☐
CMS Community Administrators	✓	✓
CMS Designers	☐	☐
CMS Desk Administrators	✓	✓
CMS Editors	☐	✓
CMS Readers	☐	✓
Everyone	☐	☐
Gold Partners	☐	☐
Live ID users	☐	☐
Not authenticated users	☐	☐
Silver Partners	☐	☐

Group pages permissions

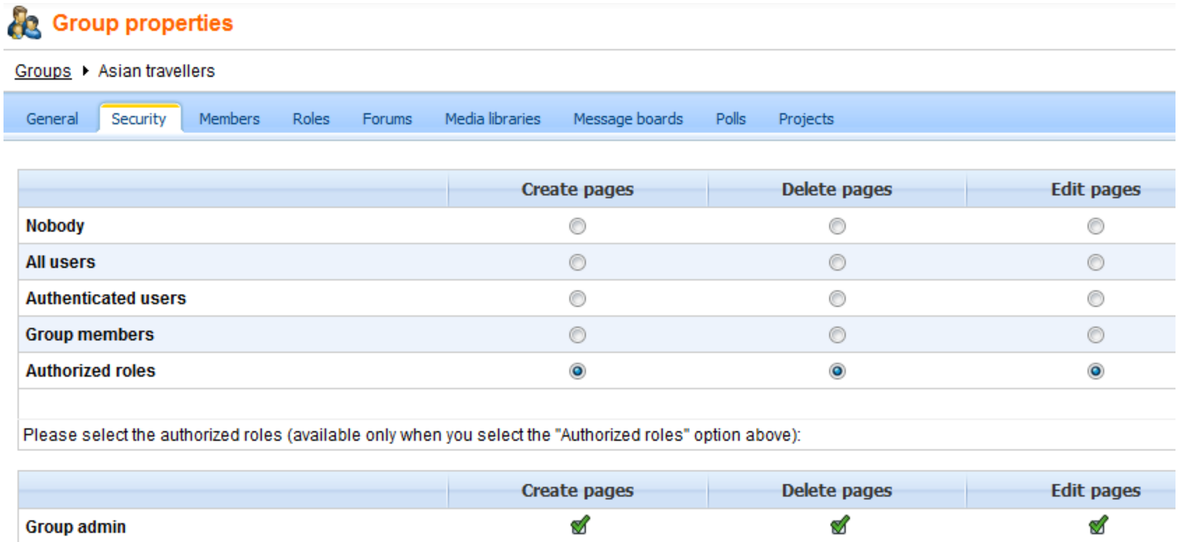
- **Create pages** - users can create group pages
- **Delete pages** - users can delete group pages
- **Edit pages** - users can edit group pages

These permissions can be assigned to:

- **Nobody** - nobody can perform the action
- **All users** - all users can perform the action
- **Authenticated users** - only signed-in users can perform the action, i.e. anonymous public users can not perform it
- **Group members** - only group members can perform the action, i.e. authenticated non-group members and anonymous users can not perform it
- **Authorized roles** - only members of the group roles selected below can perform the action

Group admin's permissions

Group administrators can perform any of these actions, even if they haven't the permissions assigned.

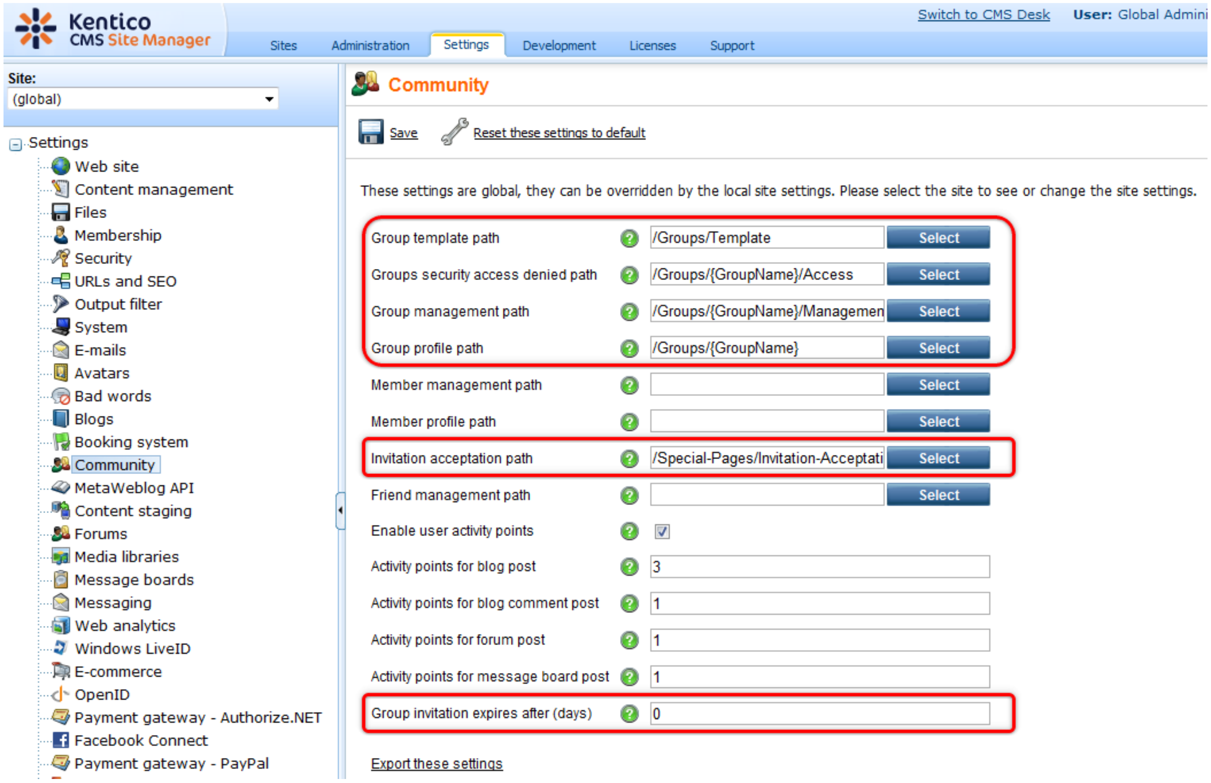


8.23.7 Settings

Settings of the groups module are located in **Site Manager -> Settings -> Community**. The following settings can be done:

- **Group template path** - alias path of the document that will be used, together with the documents stored under it, as a template for newly created groups; e.g. */Groups/Template*
- **Groups security access path** - alias path of a document to which users will be redirected when they try to access pages of a group to which they don't have permissions; this page should contain the Group security message web part; e.g. */Groups/{GroupName}/Access*
- **Group management path** - alias path of the group management page, containing the Group profile web part; e.g. */Groups/{GroupName}/Management*
- **Group profile path** - alias path of the group profile page; e.g. */Groups/{GroupName}*

- **Invitation acceptation path** - alias path of the document containing the Group invitation web part; this is a special web part handling requests for joining a group when a user clicks the joining link in group invitation e-mail; e.g. */Special-pages/Invitation-acceptation*
- **Group invitation expires after (days)** - when some user receives a group invitation e-mail, the link for joining the group included in the e-mail will be active for the number of days entered here; after the specified duration, the link will no longer be functional; when 0 is entered, the link will be functional permanently



8.23.8 Groups internals and API

8.23.8.1 Database tables and API classes

The Groups module uses the following database tables:

- **Community_Group** - contains groups and their settings
- **Community_GroupMember** - contains information about group members
- **Community_GroupRolePermission** - contains group role permissions
- **Community_Invitation** - contains information about invitations to groups

The Groups API is provided by the following **CMS.Community** namespace classes:

- **GroupInfo**, **GroupInfoProvider** - these classes provide functionality for managing groups
- **GroupMemberInfo**, **GroupMemberInfoProvider** - these classes provide functionality for managing group members
- **InvitationInfo**, **InvitationInfoProvider** - these classes provide functionality for managing group invitations

The following topics show examples of how these classes can be used:

- Getting group data
- Modifying and creating groups
- Managing group members
- Managing group roles
- Creating a group invitation

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.23.8.2 Getting group data

The following sample code shows how you can get a group as a *GroupInfo* object:

[C#]

```
using CMS.Community;

...

        string groupName = "customGroupName";
        string siteName = "CommunitySite";

        // Get group object by GroupID
        GroupInfo gi = GroupInfoProvider.GetGroupInfo(1);

        // Get group object by group code name and site name
        GroupInfo gi2 = GroupInfoProvider.GetGroupInfo(groupName, siteName);
```

The following sample code shows how you can get a DataSet containing all groups in the system with a *GroupID* greater than 5, ordered by their *Display name*:

[C#]

```
using System.Data;
using CMS.Community;

...

        string where = "GroupID > 5";
        string orderBy = "GroupDisplayName";

        // Get DataSet of group objects
        DataSet ds = GroupInfoProvider.GetGroups(where, orderBy);
```

8.23.8.3 Modifying and creating groups

The following sample code shows how you can modify the *Display name* of an existing group:

[C#]

```
using CMS.Community;

...

        // Get group object by GroupID
        GroupInfo gi = GroupInfoProvider.GetGroupInfo(1);

        // Change group display name
        gi.GroupDisplayName = gi.GroupDisplayName + "_Updated";

        // Save object
        GroupInfoProvider.SetGroupInfo(gi);
```

The following sample code shows how you can create a new group and add it to the system:

[C#]

```
using System;
using CMS.Community;
using CMS.SiteProvider;
using CMS.UIControls;
using CMS.CMSHelper;

...

        // Create empty group object
        GroupInfo gi = new GroupInfo();

        // Set properties
        gi.GroupDisplayName = "New group";
        gi.GroupName = "NewGroup";
        gi.GroupSiteID = 2;
        gi.GroupDescription = "";
        gi.GroupApproveMembers = GroupApproveMembersEnum.AnyoneCanJoin;
        gi.GroupAccess = SecurityAccessEnum.AllUsers;
        gi.GroupApproved = true;
        gi.GroupApprovedByUserID = CurrentUser.UserID;
        gi.GroupCreatedByUserID = CurrentUser.UserID;
        gi.AllowCreate = SecurityAccessEnum.GroupMembers;
        gi.AllowDelete = SecurityAccessEnum.GroupMembers;
        gi.AllowModify = SecurityAccessEnum.GroupMembers;
        gi.GroupNodeGUID = Guid.Empty;

        // Save object to database
        GroupInfoProvider.SetGroupInfo(gi);
```

8.23.8.4 Managing group members

The following sample code shows how you can add the user *gold* as a member of the group:

[C#]

```
using System;
using CMS.Community;
using CMS.SiteProvider;
using CMS.UIControls;
using CMS.CMSHelper;

...

        string groupName = "customGroupName";
        string siteName = "CommunitySite";

        // Get group object
        GroupInfo gi = GroupInfoProvider.GetGroupInfo(groupName, siteName);

        // If group exist
        if (gi != null)
        {
            // Get user object
            UserInfo ui = UserInfoProvider.GetUserInfo("gold");

            // If user exists
            if (ui != null)
            {
                // Create new group member object
                GroupMemberInfo gmi = new GroupMemberInfo();

                //Set properties of member
                gmi.MemberGroupID = gi.GroupID;
                gmi.MemberApprovedByUserID = CurrentUser.UserID;
                gmi.MemberApprovedWhen = DateTime.Now;
                gmi.MemberInvitedByUserID = CurrentUser.UserID;
                gmi.MemberUserID = ui.UserID;
                gmi.MemberJoined = DateTime.Now;

                // Save object to database
                GroupMemberInfoProvider.SetGroupMemberInfo(gmi);
            }
        }
```

The following sample code shows how you can remove the user *gold* from the group:

[C#]

```
using CMS.Community;
using CMS.SiteProvider;

...

        // Get user object
        UserInfo ui = UserInfoProvider.GetUserInfo("gold");
```

```
        // If user exists
        if (ui != null)
        {

            int userID = ui.UserID;
            int groupID = 13;

            // Get group member object by UserID and GroupID
            GroupMemberInfo gmi = GroupMemberInfoProvider.GetGroupMemberInfo(userID,

            // Delete group member object from database
            GroupMemberInfoProvider.DeleteGroupMemberInfo(gmi);

            // Delete group member object from database by GroupMemberID
            GroupMemberInfoProvider.DeleteGroupMemberInfo(1);
        }
```

The following sample code shows how you can modify the *Comment* property of a group member:

[C#]

```
using CMS.Community;

...

        // Get group member object by GroupMemberID
        GroupMemberInfo gmi = GroupMemberInfoProvider.GetGroupMemberInfo(1);

        // Change member comment property
        gmi.MemberComment = gmi.MemberComment + " Updated";

        // Save object
        GroupMemberInfoProvider.SetGroupMemberInfo(gmi);
```

The following sample code shows how you can get a DataSet containing all group members in the system with a *GroupMemberID* greater than 2, ordered by their *GroupMemberID*:

[C#]

```
using System.Data;
using CMS.Community;

...

        string where = "MemberID > 5";
        string orderBy = "MemberID";

        // Get DataSet of group member objects
        DataSet ds = GroupMemberInfoProvider.GetGroupMembers(where, orderBy);
```

8.23.8.5 Managing group roles

The following sample code shows how you can add a role to a group:

[C#]

```
using CMS.Community;
using CMS.SiteProvider;
using CMS.CMSHelper;

...

        string groupName = "customGroupName";
        string siteName = "CommunitySite";

        // Get group object
        GroupInfo gi = GroupInfoProvider.GetGroupInfo(groupName, siteName);

        // If group exists
        if (gi != null)
        {
            // Create role object
            RoleInfo ri = new RoleInfo();

            // Set role properties
            ri.RoleName = "CustomGroupRole";
            ri.DisplayName = "Custom group role";
            ri.RoleGroupID = gi.GroupID;
            ri.SiteID = CMSContext.CurrentSiteID;

            // Save role to database
            RoleInfoProvider.SetRoleInfo(ri);
        }
```

The following sample code shows how you can add a group member to a group role:

[C#]

```
using CMS.SiteProvider;

...

        string roleName = "CustomGroupRole";
        string siteName = "CorporateSite";

        // Get role object
        RoleInfo ri = RoleInfoProvider.GetRoleInfo(roleName, siteName);

        // Get user object
        UserInfo ui = UserInfoProvider.GetUserInfo("CustomGroupMember");

        // If role and user exist
```

```
        if ((ri != null) && (ui != null))
        {
            // Add user to role
            UserRoleInfoProvider.AddUserToRole(ui.UserID, ri.RoleID);
        }
```

8.23.8.6 Creating a group invitation

The following sample code shows how you can create an invitation to a group:

[C#]

```
using System;
using CMS.Community;
using CMS.UIControls;
using CMS.CMSHelper;

...

        // Create new invitation object
        InvitationInfo ii = new InvitationInfo();

        // Set properties
        ii.InvitationComment = "Invitation comment";
        ii.InvitationGroupID = 1;
        ii.InvitationUserEmail = "admin@localhost.local";
        ii.InvitedByUserID = CurrentUser.UserID;
        ii.InvitationCreated = DateTime.Now;
        ii.InvitationValidTo = DateTime.Now.AddDays(1);

        // Save object
        InvitationInfoProvider.SetInvitationInfo(ii);
```

8.24 Image gallery

8.24.1 Overview

The Image gallery module is used for effortless creating of image gallery pages. By means of this module, you can display images which are saved in the content tree as documents. The module encompasses four page templates and three web parts suitable for creating image galleries.

However, using the Image gallery module is not the only way to display images on your web pages. You can achieve similar functionality by using the Media libraries module. The Media libraries module provides an alternative way of creating a gallery and is suitable if large numbers of stored images are to be displayed or if you want to make audio, video and other file types accessible from your web pages. Another option consists in attaching an image to a document using the Attachment image gallery web part.

In the Image gallery chapter, you can learn what web parts are available in the Image gallery module (learn here how to do it), what page templates can be used (as explained here), you can learn how to import files or even the whole folder structure from the disk to Kentico CMS content repository (as referenced here) and you can also learn how to use transformations to alter the appearance of your gallery (as explained here).

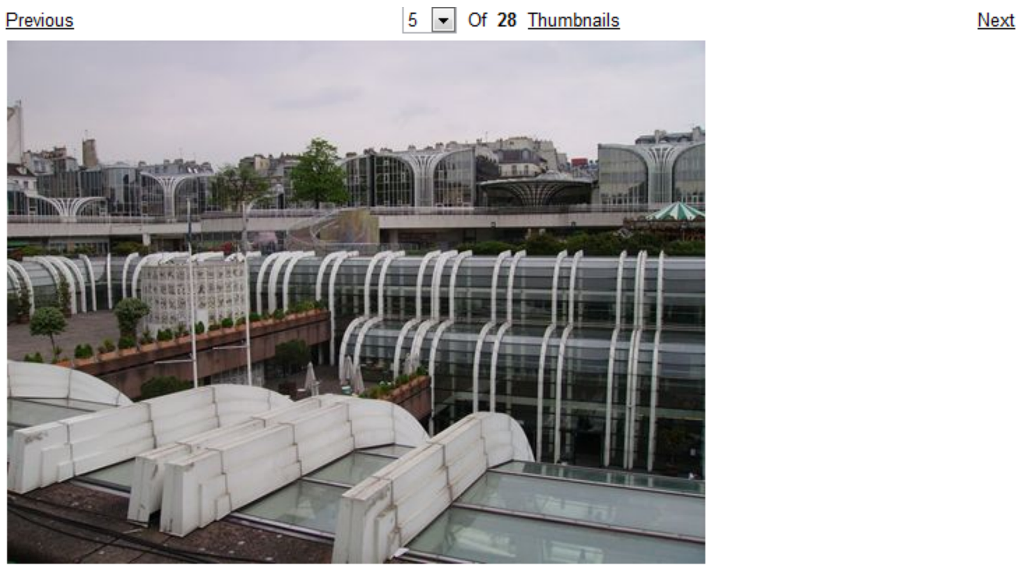
8.24.2 Available web parts

There are three web parts suitable for creating image galleries. From the **Select web part** dialog window, you can select the **Image gallery**, **Lightbox gallery** and **Content slider** web parts in the **Listings and viewers** category.

Image gallery

This is the basic web part for image galleries. In its initial view, it displays a set of picture thumbnails:

After clicking one of the thumbnails, the detail view will be displayed:



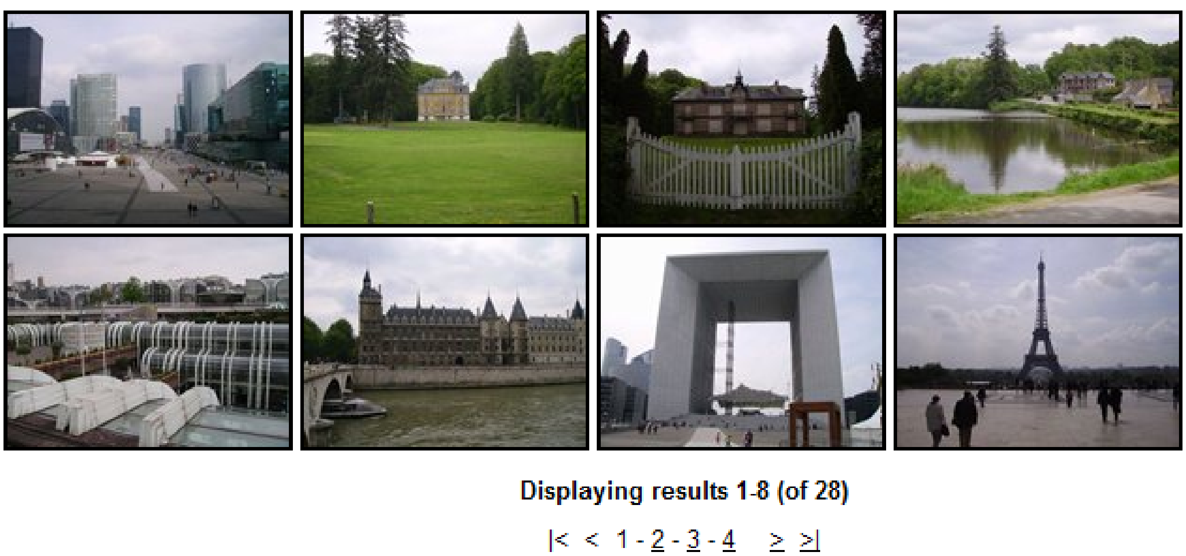
Besides the usual parameters common to all web parts, these properties can be set to customize the appearance of the gallery:

Transformations	
Detail transformation	Transformation used for displaying a selected image.
Thumbnail transformation	Transformation used for displaying gallery thumbnails.
Layout	

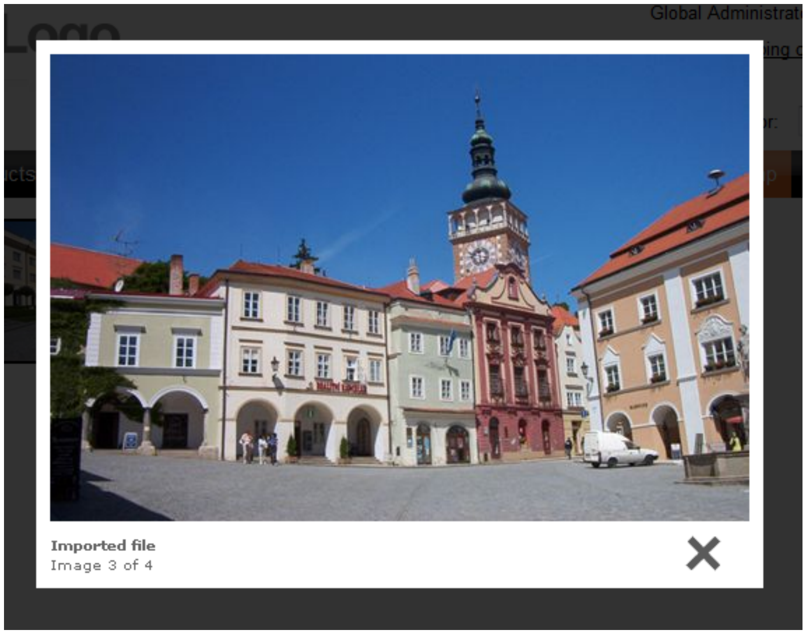
Number of columns	Number of thumbnail columns in the thumbnail view.
Rows per page	Number of thumbnail rows per page in the thumbnail view.
Paging	
Paging mode	Paging parameter transfer type: Query string - the paging parameter is transferred through URL Postback - the actual page is transferred through ViewState, no URL parameter is used
Query string key	Name of the URL parameter containing the page number.
Show first and last buttons	If checked, buttons leading to the first and last pages of the gallery will be displayed.
Show buttons on top	If checked, paging buttons will be shown above the thumbnails. Otherwise, they will be displayed below them.

Lightbox gallery

This web part's thumbnail view is similar to that of the **Image gallery** web part:



After clicking one of the thumbnails, the whole page will be grayed out and a lightbox with the selected image will be displayed on the top of it, as you can see in the screenshot below:



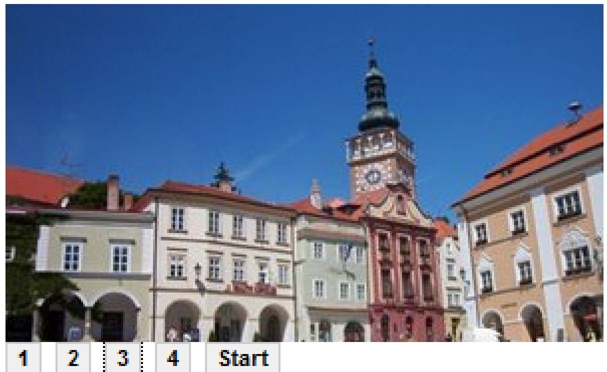
Here is a list of properties specific to the **Lightbox gallery** web part:

Transformations	
Transformation	Transformation used for displaying the list of thumbnails.
Alternating transformation	Transformation used for even items in the thumbnail view.
Selected item transformation	Transformation used in the detail view mode.
Item separator	Separator displayed between thumbnails.
Nested controls ID	Sets the nested controls IDs. Use ';' as a separator; Example: myRepeaterID;myDatalistID;myRepeaterID2 This property replaces the previously used NestedRepeaterID and NestedDataListID properties. If you are still using these properties, no changes to functionality will occur, but it is advisable to rewrite your code to use the new property instead.
Paging	
Enable paging	Indicates if paging is enabled. If unchecked, all thumbnails in the gallery will be displayed on a single page.
Paging mode	Paging parameter transfer type: Query string - the paging parameter is transferred through URL Postback - the actual page is transferred through ViewState, no URL parameter is used
Pager position	Determines position of the pager. Available options are *Bottom*, *Top* and *Top and bottom.*
Page size	Number of thumbnails displayed per page.

Query string key	Name of the URL parameter containing the page number.
Show first and last buttons	If checked, buttons leading to the first and last page of the gallery will be displayed.
LightBox Configuration	
Always visible navigation buttons	Indicates whether the navigation buttons are always visible, not only on mouse over.
Frame width	Width of the LightBox frame.
Frame height	Height of the LightBox frame.
Path to external CSS file	URL path to the external CSS file required by LightBox.
Overlay opacity	Opacity of LightBox background. Enter values ranging from 0 (transparent) to 1 (opaque black).
Animate	Enables LightBox animation.
Load delay	Load delay time (in miliseconds). If you are using automatic resizing, this value indicates how long the lightbox will take to reach the element size. If you have problems with displaying lightbox content, try to use a higher value.
Resize speed	Defines the speed of resizing images. Choose values ranging from 1 (slowest) to 10 (fastest).
Border size	Size of the image border.
Loading image	Image displayed while loading the LightBox image.
Close button image	Image of the Close button.
Previous button image	Image of the Previous button.
Next button image	Image of the Next button.
Group name	Name of the LightBox group.

Content slider

The **Content slider** is a web part that can be used for displaying various document types, hence it is also very suitable for displaying images. Contrary to the previous two web parts, the **Content slider** provides no thumbnail view. It displays a full sized image slide show with a pager below. The pager allows for browsing through the images using the numbered buttons. After clicking any of these buttons, the slide show stops and the **Start** button appears. This button launches the slide show again.



Specific properties of the **Content slider** web part:

Transformations	
Transformation	Transformation used for displaying the list of thumbnails.
Alternating transformation	Transformation used for even items in the thumbnail view.
Item separator	Separator displayed between thumbnails.
Nested controls ID	Sets the IDs of nested controls as specified in the transformation code.
Div options	
Width (px)	Width of the scrolling text area in pixels.
Height (px)	Height of the scrolling text area in pixels.
Style	Style assigned to the DIV tag of the area.
JavaScript options	
FadeIn time (milliseconds)	Fade in time of the image.
FadeOut time (milliseconds)	Fade out time of the image.
Break time (milliseconds)	Time for which the image will be displayed.
Auto start	If checked, the slide show will automatically start from the beginning.

8.24.3 Available page templates

There are three basic page templates that can be used for image galleries. In **CMS Desk -> Content**, create a new **Page (menu item)** using a page template. From the **Use page template** dialog window, you can select the **Image gallery**, **Lightbox gallery**, **Sliding gallery** and **List of galleries** templates in the **Images** category.

List of galleries

This page template is used for displaying a list of all galleries under a selected path. For each gallery, it displays a thumbnail with a gallery name above it. By clicking one of the thumbnails, you will be redirected to the main page of the gallery.

The **Teaser image** of each gallery's menu item is used as the thumbnail in the list of galleries. To change some of the thumbnails, select the appropriate gallery's menu item in the content tree and switch to **CMSDesk -> Edit -> Form**. On the displayed page, select a new **Teaser image** and click **Save**. The selected image will now be displayed as a thumbnail of the gallery in the list of galleries.

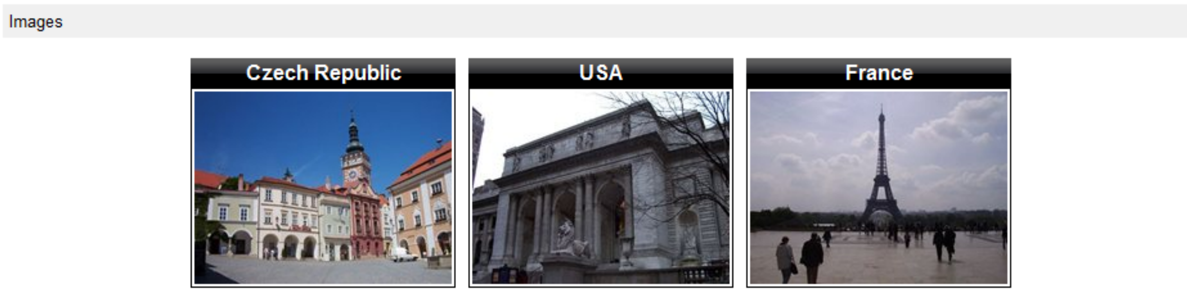


Image gallery

This is a basic page template used for displaying image galleries. It uses the **Image gallery** web part for displaying images under a given path in the content tree. See the Available web parts chapter for detailed info.

Lightbox gallery

This page template uses the **Lightbox gallery** web part for displaying images. See the Available web parts chapter for detailed info.

Sliding gallery

Displays images using the **Content slider** web part.

List of galleries

Displays list of galleries.

8.24.4 Importing images

Images used in galleries are imported the same way as any other files, as described in the File import chapter.

8.24.5 Transformations

Use of transformations is essential for all **Image gallery** web parts. You can view and alter transformations in the **Transformations** section of a web part properties window. For each transformation, you can **Select** a predefined transformation from a list, **Edit** the current transformation or create a **New** one by means of the respective buttons.

Here are some examples of how you can alter the appearance of the **Image gallery** web part. This is the default thumbnail transformation code of the **Image gallery** web part:

```
<a href="?imagepath=<%# System.Web.HttpUtility.UrlEncode(DataBinder.Eval
(Container, "DataItem.NodeAliasPath").ToString()) %>">
<%# IfEmpty(Eval("FileAttachment"), "no image", "<img alt=\"" + Eval("FileName") +
"\" src=\"" + GetFileUrl("FileAttachment") + "?maxsidesize=180\" border=\"0\" /
>") %>
</a>
```

And this is how the gallery created using this transformation looks like:

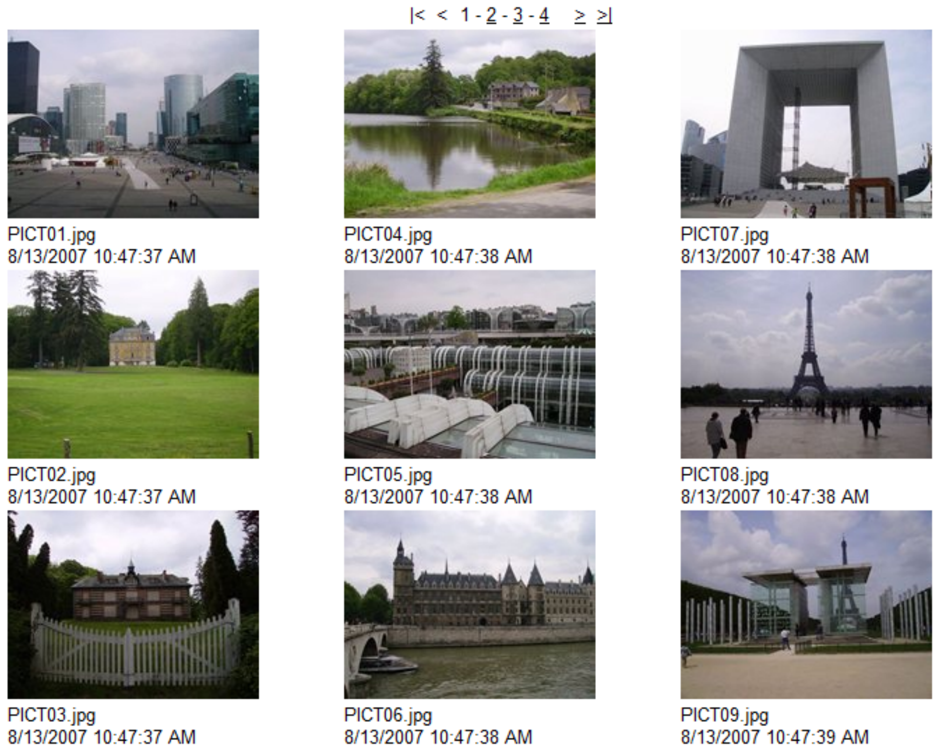
In the following example, we will add a red border around each thumbnail.

```
<div style="border: solid 3px Red">
<a href="?imagepath=<%# System.Web.HttpUtility.UrlEncode(DataBinder.Eval
(Container, "DataItem.NodeAliasPath").ToString()) %>">
<%# IfEmpty(Eval("FileAttachment"), "no image", "<img alt=\"" + Eval("FileName") +
"\" src=\"" + GetFileUrl("FileAttachment") + "?maxsidesize=180\" border=\"0\" /
>") %>
</a>
</div>
```

This is how the result looks like:

This example shows how to display the file name and date and time of creation for each thumbnail in the gallery:

```
<a href="?imagepath=<%# System.Web.HttpUtility.UrlEncode(DataBinder.Eval
(Container, "DataItem.NodeAliasPath").ToString()) %>">
<%# IfEmpty(Eval("FileAttachment"), "no image", "<img alt=\"" + Eval("FileName") +
"\" src=\"" + GetFileUrl("FileAttachment") + "?maxsidesize=180\" border=\"0\" /
>") %>
</a>
<%# Eval("DocumentName") %> <br/>
<%# GetDateTime("DocumentCreatedWhen") %>
```



Transformations for the Lightbox gallery web part

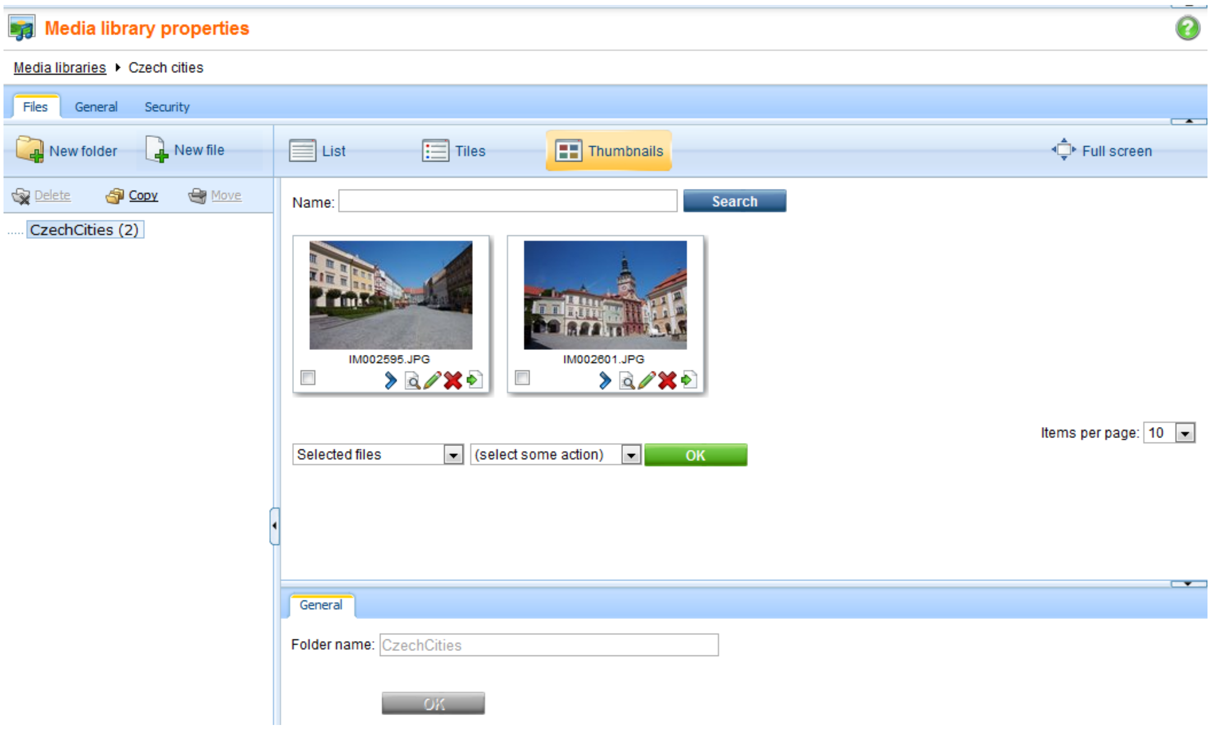
When writing a custom transformation for the **Lightbox gallery** web part, it is necessary to use the 'rel' and 'rev' parameters as highlighted in the transformation code below. The 'title' parameter is used to determine the description of the image displayed in the lightbox.

```
<a href="<%# GetDocumentUrl() %>" rel="lightbox[group]" rev="<%# Eval
("NodeAliasPath") %>"
title="<%# Eval("FileDescription") %>"><img src="<%# GetFileUrl
("FileAttachment") %>?maxsidesize=150"
alt="<%# Eval("FileName") %>" /></a>
```

8.25 Media libraries

8.25.1 Overview

The Media libraries module enables storing of various files, such as photos, sound, videos, package files, presentation files, etc. This means that not only media files, but also other types of files can be stored in media libraries. Media libraries can either be global or related to a particular group.



Main features of the Media libraries module:

- Support for storage of large amount of files (photo galleries)
- Support for storage of large files (video files)
- Each group of users can have any number of libraries
- Files can be uploaded to the libraries externally, e.g. using FTP

However, the Media libraries module is not the only way to store files of various kinds within the system. Kentico CMS allows you to upload files (such as GIF, JPG, SWF, PDF, XLS, DOC, etc.) to the Kentico CMS database or file system and manage them as any other content as described in here. Another way to achieve similar functionality is to make use of the Image gallery module (as explained here). Finally, you can attach files directly to documents and thus bind them to a document's life cycle (as explained here). Please note that both image galleries and document attachments can only be used with image file types.

- If you would like to learn how to create media libraries, please refer to the Creating media library topic.
- If you would like to learn how to upload files into them, please refer to the Uploading files topic.
- If you would like to learn how to manage files stored in a media library, please follow the Files and folders management topic.
- If you would like to read a brief overview of supported file types that can be played or displayed on your site by default, please refer to the Supported file types topic.
- If you would like to learn about the supported file size in media libraries, please refer to the Supported file size topic.
- If you would like to learn about names of files uploaded to a media gallery and their thumbnails, please refer to the File naming conventions topic.
- If you would like to learn how to display media gallery content on your site, please follow the Using Media library web part topic.
- If you would like to learn about integration of media library files into text using the built-in WYSIWYG

editor, please refer to the Using WYSIWYG editor topic.
- If you would like to adjust the settings of the Media libraries module, please refer to the Media library settings topic.
- If you would like to learn how to customize the location of all libraries of a particular site, please refer to the Configuring custom storage for media library topic.
- If you would like to read a more detailed explanation of how to enable any media types to be recognized by the system, please refer to the Configuring custom file types topic.
- If you would like to learn how to configure maximal uploaded file size, please refer to the Configuring maximal uploaded file size topic.
- If you would like to learn how to deal with secured and non-secured libraries on your site, please refer to the Secured vs. Non-secured libraries topic.
- If you would like to learn how to set media library permissions, please refer to the Media library permissions topic.

Please note that media library files are stored directly in the file system, organized in folders. The default location of all libraries of a particular site is **~/<sitename>/media**. Each media library is either global in the context of a particular site or belongs to a group.

More examples of the use of the Media libraries module are available in Kentico CMS Community Site Guide; please note that these practical examples do not concern the whole functionality of the module but focus on its use in a broader context of the Community Site sample website:

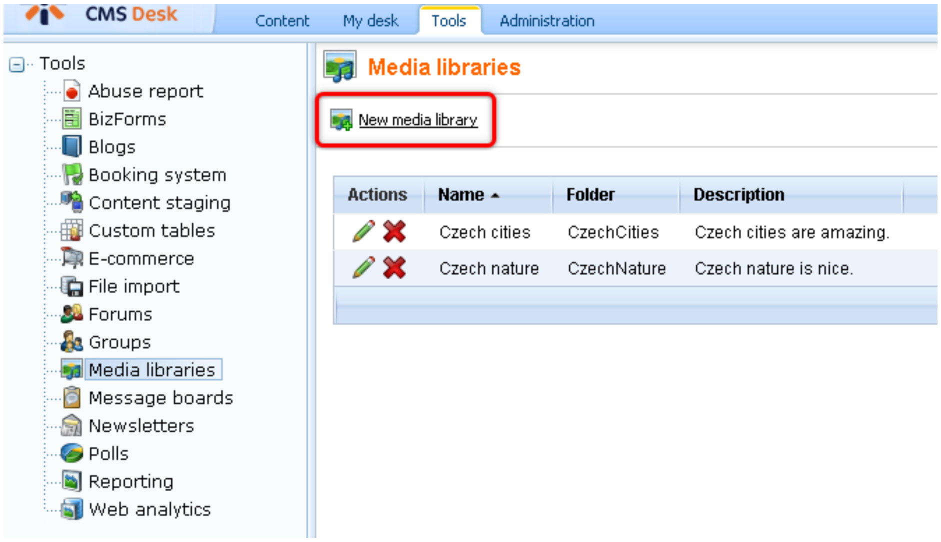
- See Part 1 -> Media libraries -> Current functionality in Kentico CMS Community Site Guide: A brief description of the functionality of the Media libraries module.
- See Creating custom media libraries in the same section of the Community Site Guide: An example of creating a new global media library and publishing it on the site.
- See Publishing more than one global media library in the same section of the Community Site Guide: An example of how to publish more than one global media library on the site.
- See Community Site Guide -> Part 2 -> Pre-development tasks -> Creating a sample Media library in Kentico CMS Community Site Guide: An example of creating a sample media library.
- See Creating the Blogs section -> Creating the Media page in the same section of the Community Site Guide: An example of creating the Media page using the **Media gallery** web part under the Blogs section.
- See Creating the Groups section -> Creating the Media page in the same section of the Community Site Guide: An example of creating the Media page using the **Media gallery** web part under the Groups section.

8.25.2 Creating media library

There are two ways how media libraries can be created:

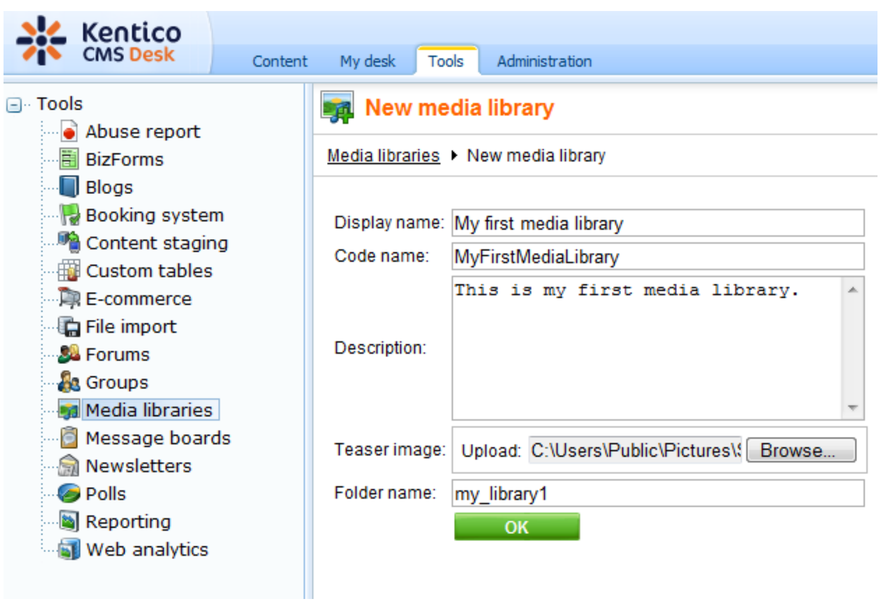
- **Live site**: group administrators can create group media libraries on the live site using the **Group profile** web part
- **Administration interface**: site administrators can create global media libraries in **CMS Desk -> Tools -> Media libraries** or group media libraries in **CMS Desk -> Tools -> Groups -> Edit () -> Media libraries**

1. In all of these cases, media libraries can be created by clicking the **New media library** link above the list of libraries, as you can see in the screenshot below.

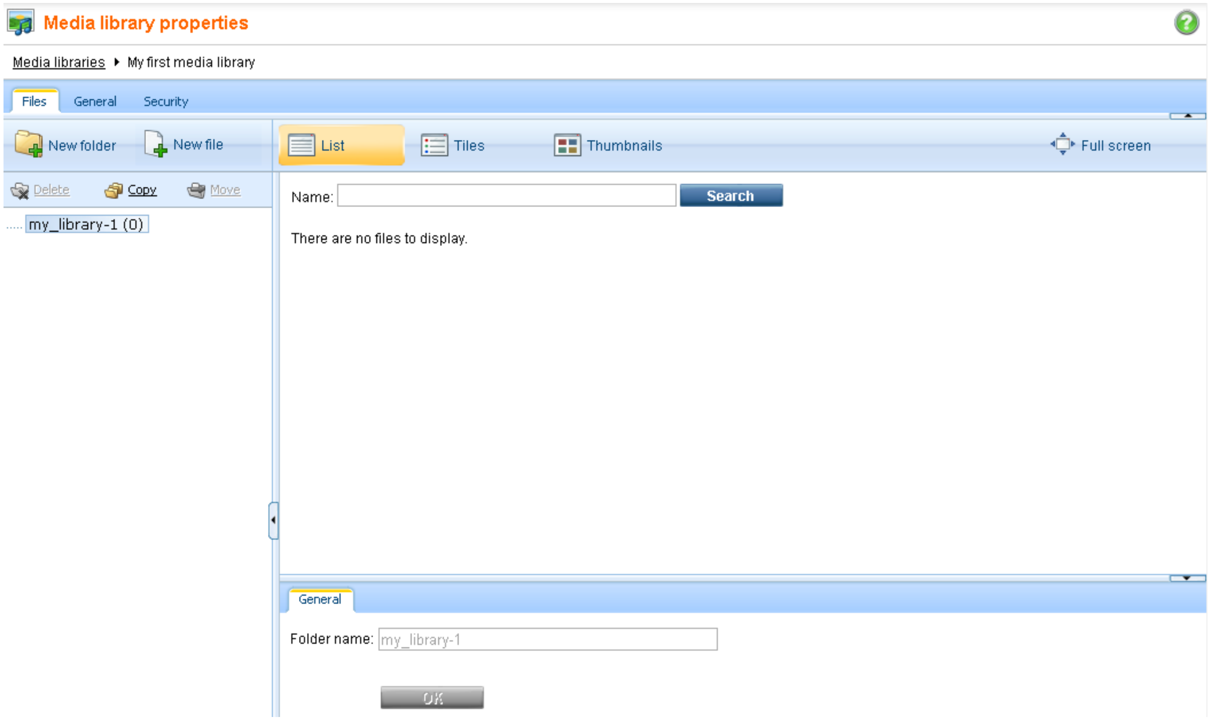


2. After clicking the link, the following details need to be entered:

- **Display name** - name of the media library displayed in the administration interface and on the live site
- **Code name** - name of the media library used in site code
- **Description** - text describing the media library
- **Teaser image** - image used as the media library teaser
- **Folder name** - name of the folder where files will be stored; this folder will be created under **<web project>\<site name>\media** or a custom location set in **Site Manager -> Settings -> Media libraries -> Media libraries folder**



3. Click **OK** to create the library. After doing so, you will be redirected to the media library's editing interface - just as if you clicked the **Edit** ( ) icon in the list of libraries.



8.25.3 Media library content

8.25.3.1 Uploading files

There are three ways how files can be uploaded into a media library:

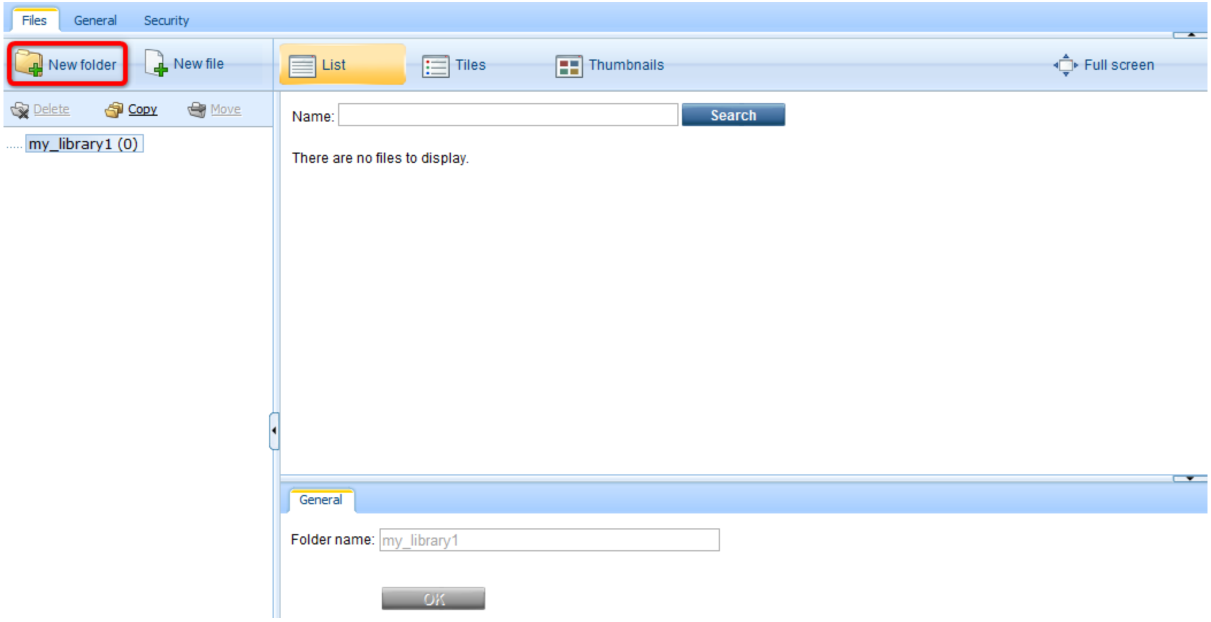
- **By means of the administration interface**
- **On the live site using the Media gallery or Media file uploader web parts**
- **Externally, directly into the file system - e.g. via FTP**

In the following examples, you will learn how to add files into media libraries via the administration interface, on the live site and using FTP.

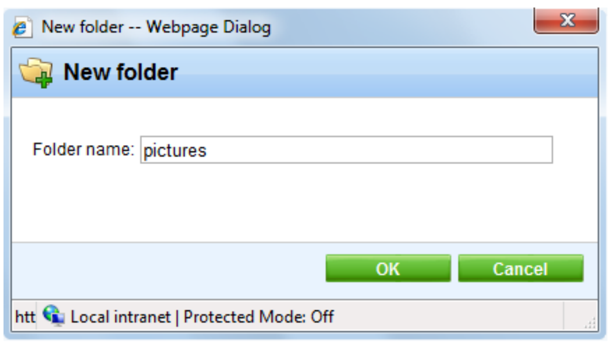
Uploading files via the administration interface

If you choose to **Edit** ( ) a media library in **CMS Desk -> Tools -> Media libraries**, you can manage files in the library on the **Files** tab.

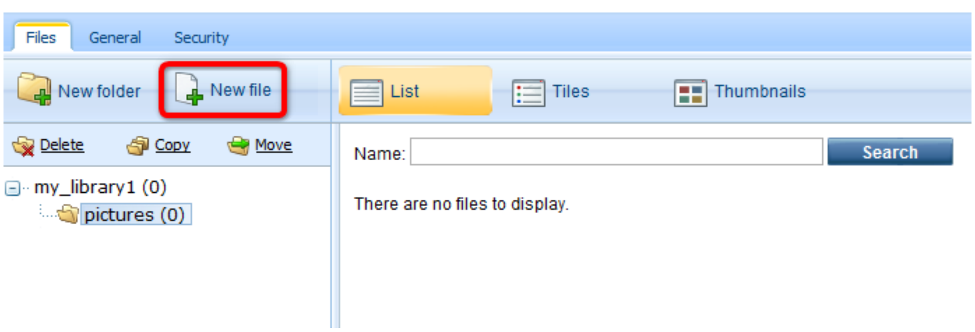
1. It is a good idea to keep your files organized in folders within the media library. To create a new folder, click the **New folder** button in the top left corner of the page.



2. Enter the name of the folder, e.g. *pictures*, and click **OK**.



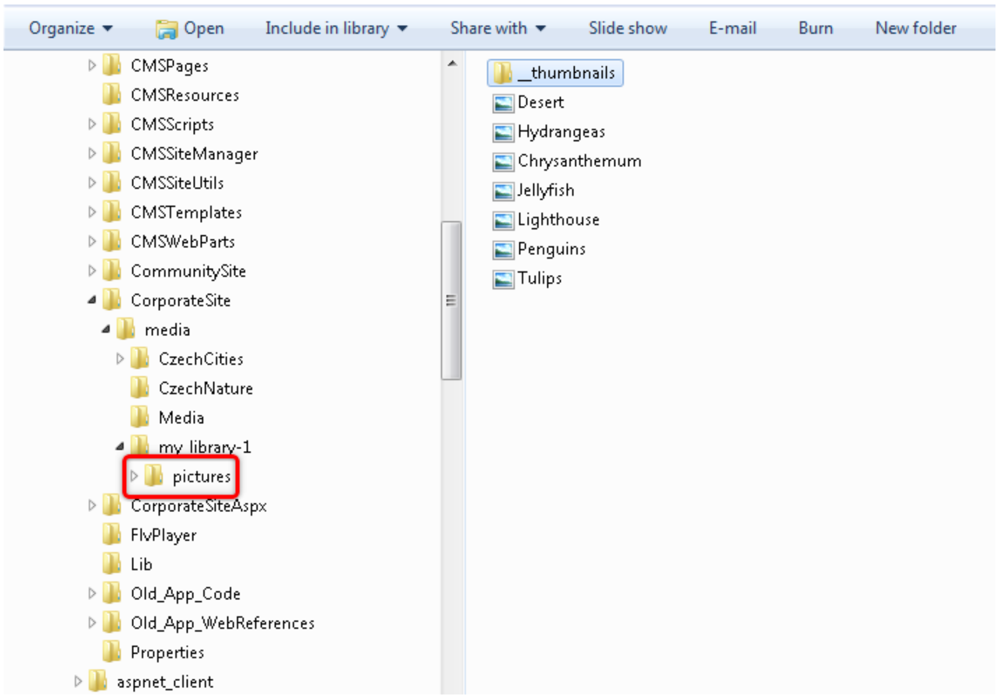
3. The folder is now created. You can upload files into the selected folder by clicking the **New file** button.



4. A dialog appears, letting you upload the file. Just select the right folder and the file to be uploaded, then click the **Open** button.

5. Now if you go to the site folder in your file system, you will find the media library under the **media** folder, as you can see the screenshot below. The location of the folder may be customized as

described here.

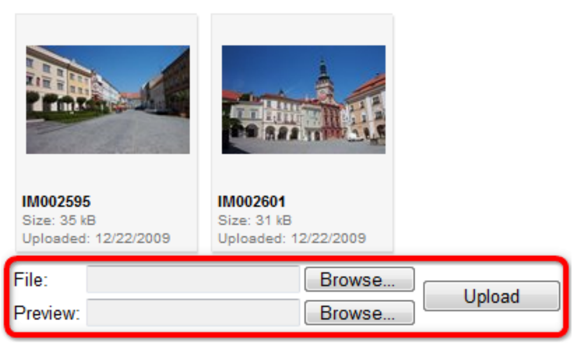


On-site upload via the Media gallery and Media file uploader web parts

The **Media gallery** web part has the following two properties enabling on-site file upload:

- **Allow upload -** enables on-site upload of files
- **Allow upload thumbnail** - enables on-site upload of file previews

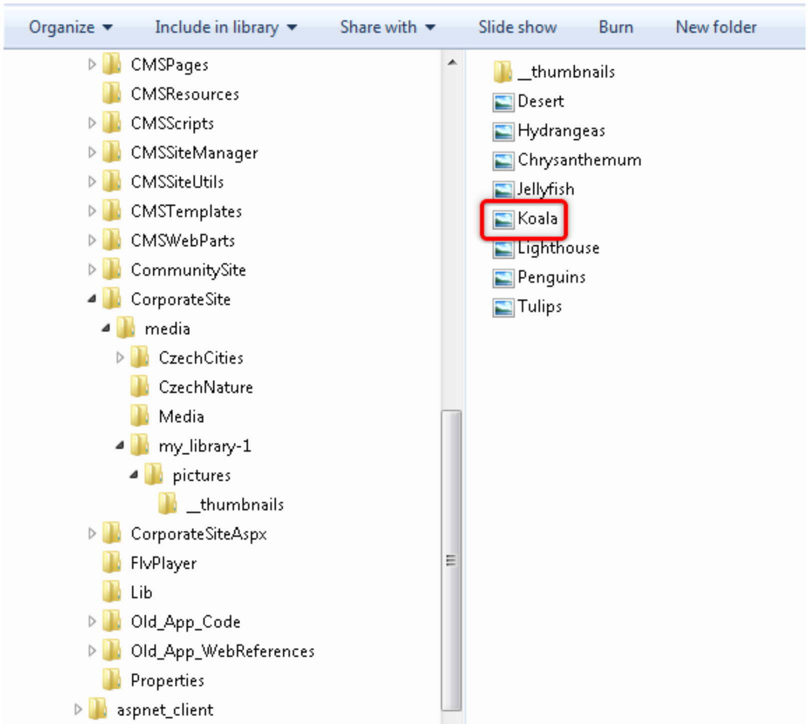
If these properties are enabled, the controls highlighted in the screenshot below will be displayed in the web part, letting users upload files and thumbnails:



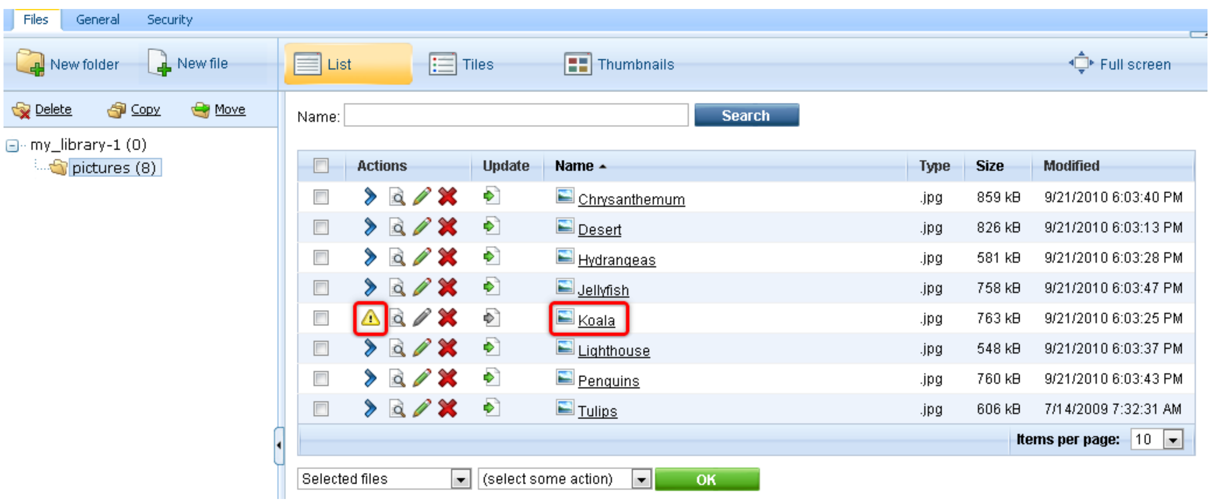
External upload

Files can also be uploaded **externally**, without any use of Kentico CMS administration interface or on-site web parts, e.g. using **FTP**. To do this, you simply need to copy the files into the appropriate folder of the media library.

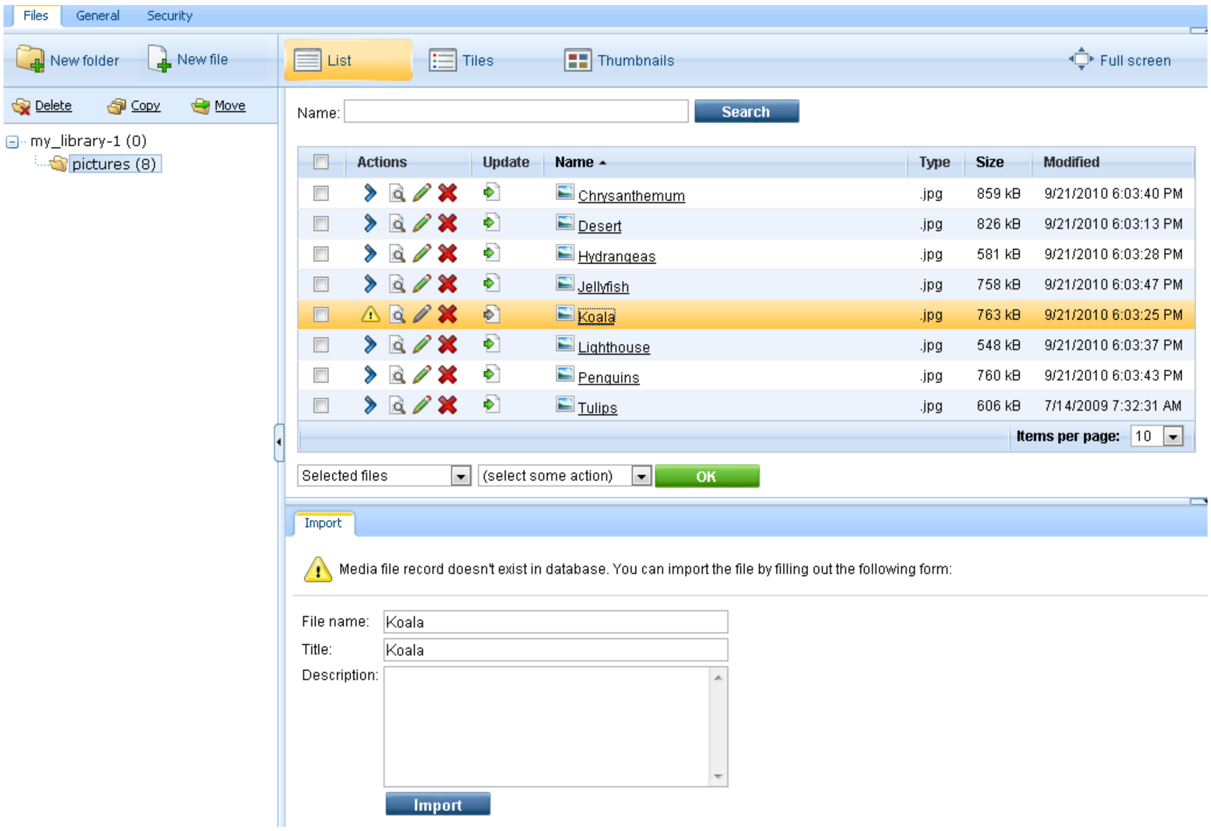
1. For the purposes of this example, try uploading another file into the **pictures** folder by copying it using Windows Explorer or any other file manager.



2. If you go back to the Media library administration interface, you will see the file present in the list. As you can notice, this file is marked with the warning (⚠) icon indicating that the file has not been imported yet. This means that the file **has not yet been registered in the database**. Such files can not be used on the site and have to be registered first. This happens only to externally uploaded files, files uploaded via the administration interface are registered in the database automatically.



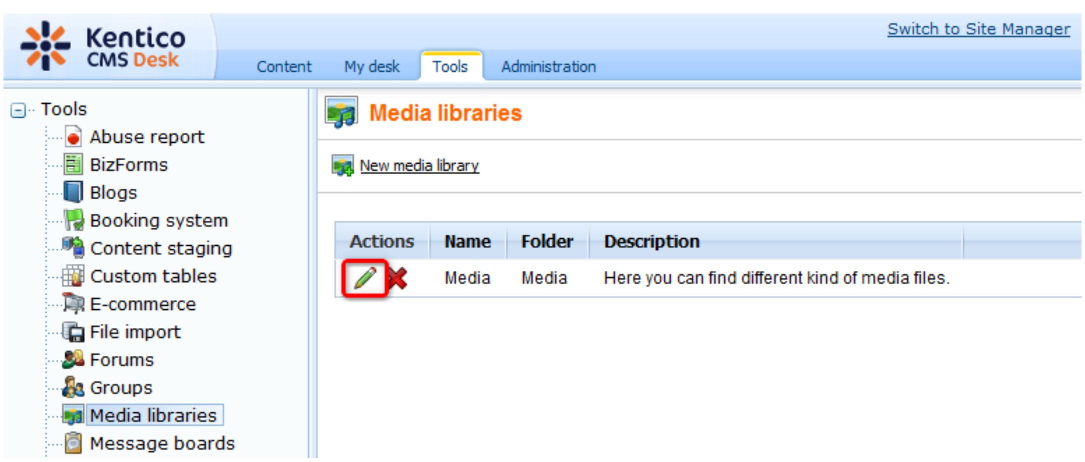
To register the file, you have to click the **Import** button, which creates the file's record in the database. Please note that the files are not physically copied into the database, they remain in the file system only and a record linking to the physical file is created in the database.



Multiple files can also be imported conveniently as a batch. See Batch operations in the **Files and folders management** topic for more details.

8.25.3.2 Files and folders management

To manage files stored in a media library, you need to go to **CMS Desk -> Tools -> Media libraries** and click the **Edit** (✎) icon next to the appropriate media library.



The library editing interface will be displayed on the **Files** tab where the files can be managed.

Folder operations

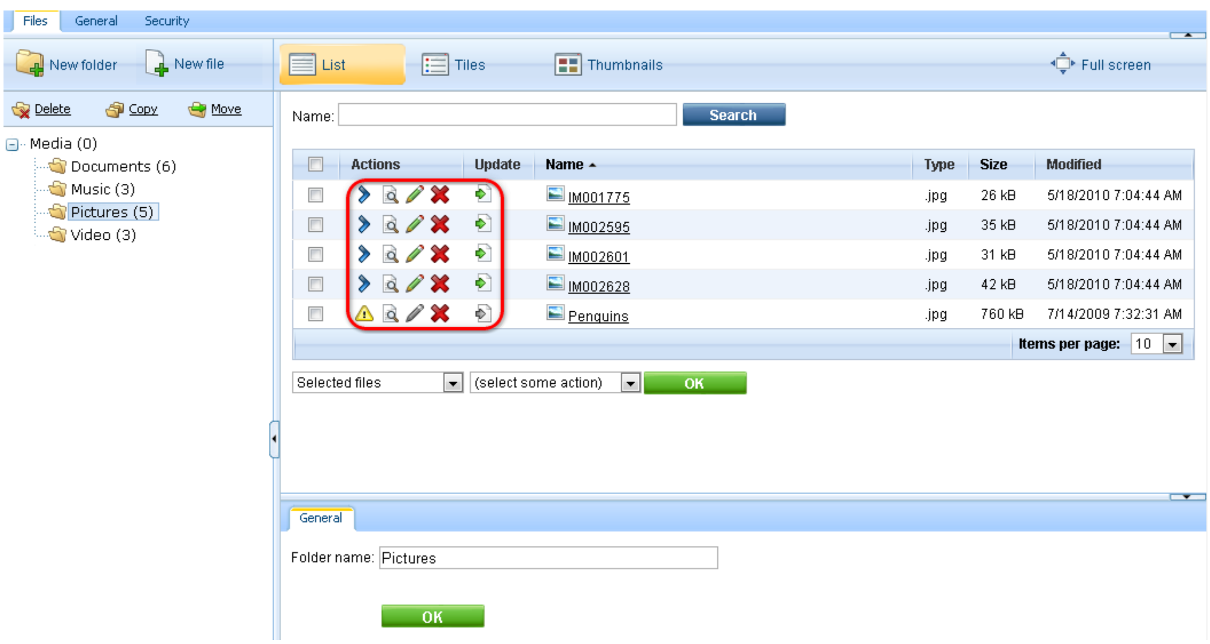
The following actions can be performed with the folders in the media library:

- **New folder** - creates a new folder under the currently selected folder
- **Delete** - deletes the currently selected folder
- **Copy** - uploads the folder and its content to another location and keeps the original folder
- **Move** - uploads the folder into another location and deletes the original folder

Single file operations

You can perform the following operations by clicking the icons next to particular files:

- **New file** - uploads a file from the local or network file system to the currently selected folder
- **Select** - selects the file and opens its editing interface in the bottom section
- **View** - downloads the file and opens it in your default application for the file type
- **Edit** - this icon appears only with images and when clicked, the image gets opened in the built-in image editor
- **Delete** - deletes the file
- **Update** - replaces the file with another file; the original file gets deleted and replaced by the new one
- **Import** - this icon is displayed only with files that have been uploaded to the library folder externally (e.g. via FTP) and have not yet been registered in the database; by clicking the icon, you register the file in the database as described here



Batch file operations

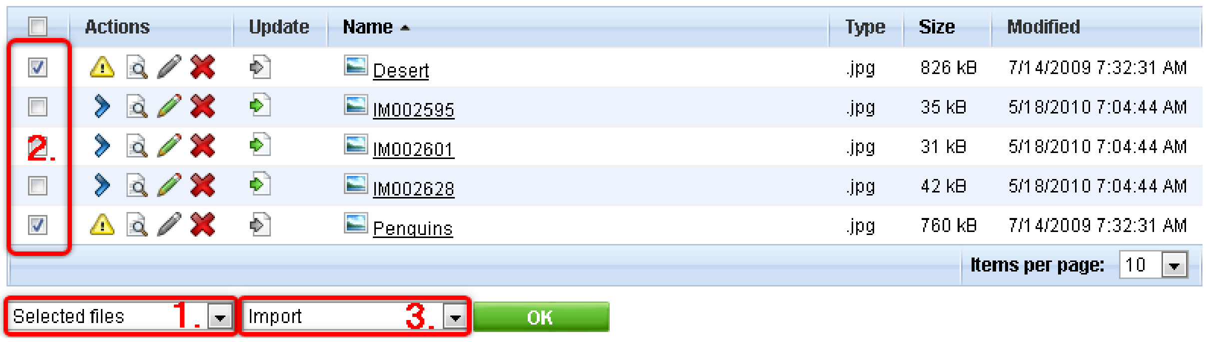
You can also perform the **Copy**, **Move**, **Delete** and **Import** operations for multiple files at once. To perform a batch operation, take the following steps:

1. Use the first drop-down list (**1.** in the screenshot) to select if you want to perform the operation for:

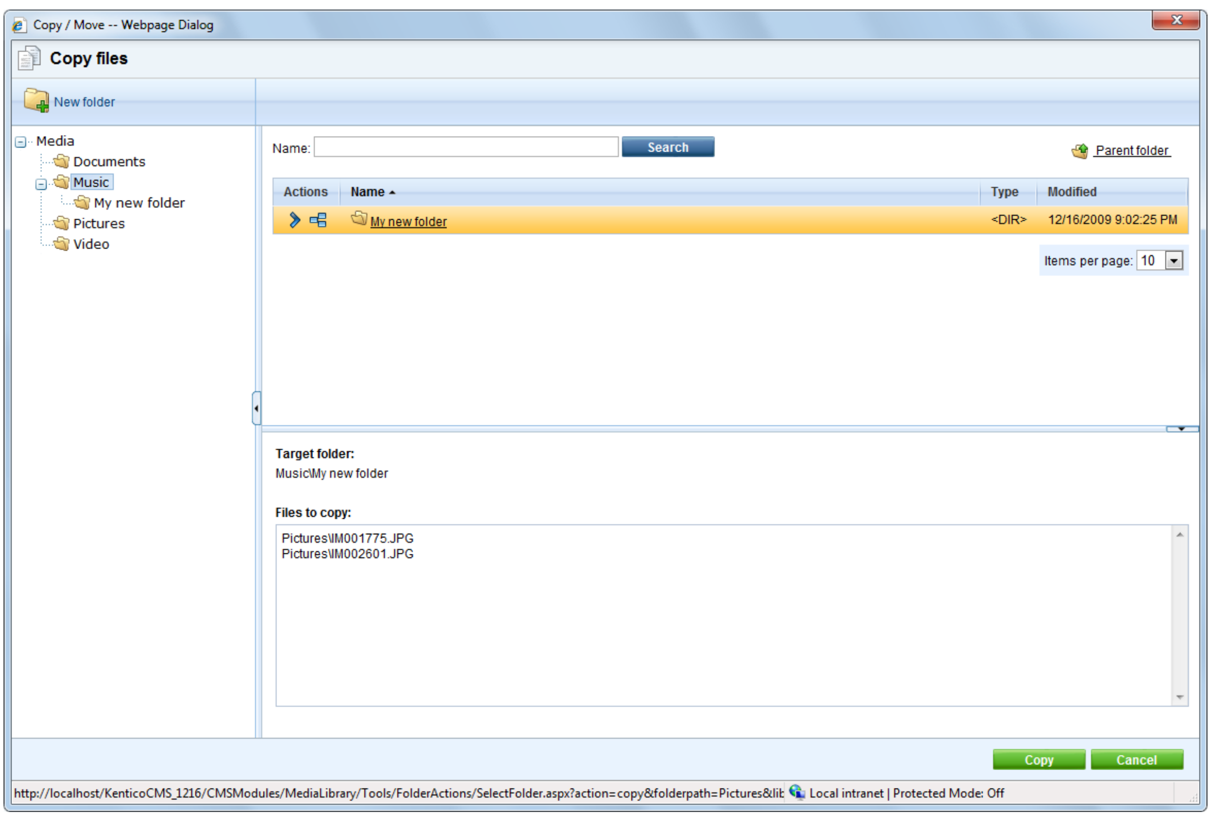
- **All files** - performs the operation for all files in the current folder
- **Selected files** - performs the operation only for files selected by the check-boxes (☑)

2. If you selected **Selected files** in the previous step, use the check-boxes (**2.** in the screenshot) to select the files to perform the operation with.

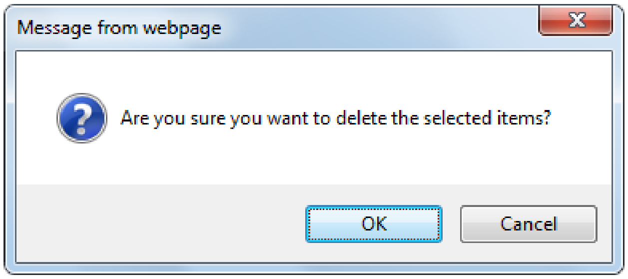
3. Choose the operation to perform using the second drop-down list (**3.** in the screenshot) and click **OK**.



4a. If you are performing the **Copy** or **Move** operation, the following dialog is displayed. Choose the target folder (or create a new one using the **New folder** button if needed) and click the **Copy/ Move** button.



4.b If you are performing the **Delete** operation, you will be prompted to confirm the deletion. Click **OK** to delete the files.

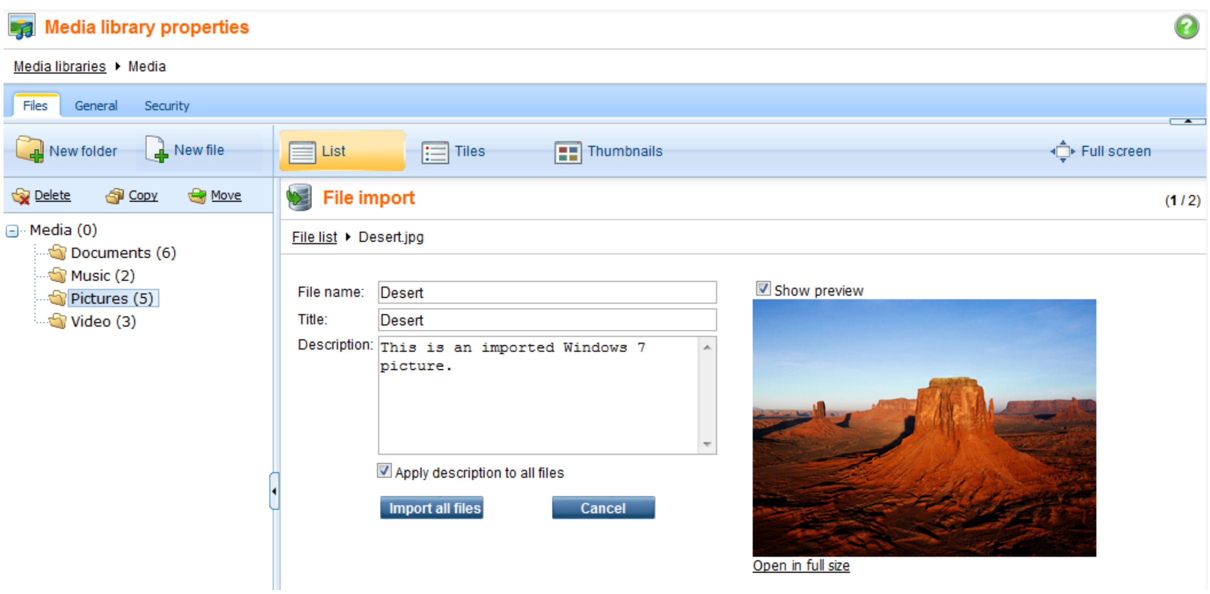


4.c If you are performing a batch **Import** operation, the dialog as in the following screenshot will appear for each imported file. You need to specify the following details for each imported file:

- **File name**: name of the file displayed in the administration interface
- **Title**: name of the file displayed on the live site
- **Description**: text describing the image

Click the **Import file** button to proceed to the following file.

You can also enable the **Apply description to all files** option, which imports all files in one click on the **Import all files** button. All files will have the entered **Description** and the **File name** and **Title** fields will contain the names of the physical files without extension.



Please note

Because of ASP.NET architecture, **site restart** will occur when:

- a media library is deleted
- a group containing a media library is deleted
- one of the following actions is performed when editing a library in **CMS Desk -> Tools -> Media libaries -> Files** or on the live site:

- folder is deleted
- folder is renamed
- folder is moved
- large number of files is deleted (100 by default, this can be set in the <*system.web*> section of your web.config by the following key: <*compilation debug*=*"true" numRecompilesBeforeAppRestart*=*"100"*>)

Because of this, it is highly recommended to allow performing of these actions only to system administrators or to the least possible number of users. The recommended practice is for the site administrators to pre-define the folder structure of the libraries when they are created and not to allow users to further modify it.

8.25.3.3 Supported file types

The following file types are supported and can be played or displayed on your site by default:

- **Images**: bmp, gif, png, wmf, jpg, jpeg, tiff, tif
- **Audio**: wav, wma, mp2, mp3, mid, midi, mpga
- **Video**: avi, mp4, mpg, mpeg, wmv, qt, mov, rm
- **Flash**: swf

Custom file types can also be defined, as described in this chapter.

Please note

All other file types will be recognized as **documents**, which means that they can not be played as videos, displayed as pictures, etc., but they can still be stored in the library and downloaded by users on the live site.

8.25.3.4 Supported file size

If you would like to learn how to configure maximal uploaded file size, please refer to this topic.

8.25.3.5 File naming conventions

When a file is being uploaded to a media library, users can upload the file itself and its thumbnail. The **thumbnail** will be displayed in the file list in the **Media gallery** web part. Thumbnails are typically used if you want to upload some non-image file and attach an image which will be displayed as the file's thumbnail in the file list. For image files, the thumbnail can but doesn't have to be done, as the image itself will be used for this purpose.

When the thumbnail image or the image itself is larger than the required thumbnail image size, its resized copy will be created with the required size. Both thumbnails and resized copies are stored in a **hidden folder** in the root of the media library folder. Name of the hidden folder can be set in **Site**

Manager -> Settings -> Media libraries -> Media file hidden folder.

Format of media library files on the disk

- **File**: <file name>.<file extension>
- **Resized file**: <hidden folder>/<file name>_<file extension>_<width>_<height>.<extension>

- **Thumbnail**: <hidden folder>/<file name>_<file extension><thumbnail suffix>.<thumbnail extension>
- **Resized thumbnail**: <hidden folder>/<file name>_<file extension><thumbnail suffix>_<width>_<height>.<thumbnail extension>

Example

If you upload the following files:

- **Uploaded file**: MyImage.jpg
- **Uploaded thumbnail**: MyPhoto.bmp

with the following settings defined in **Site Manager -> Settings -> Media libraries**:

- **Media file hidden folder**: __thumbnails
- **Media file thumbnail suffix**: _preview

the uploaded files will be stored in the following locations in the media library folder, with the following names:

- **File**: MyImage.jpg
- **Resized file**: __thumbnails/MyImage_jpg_100_200.jpg

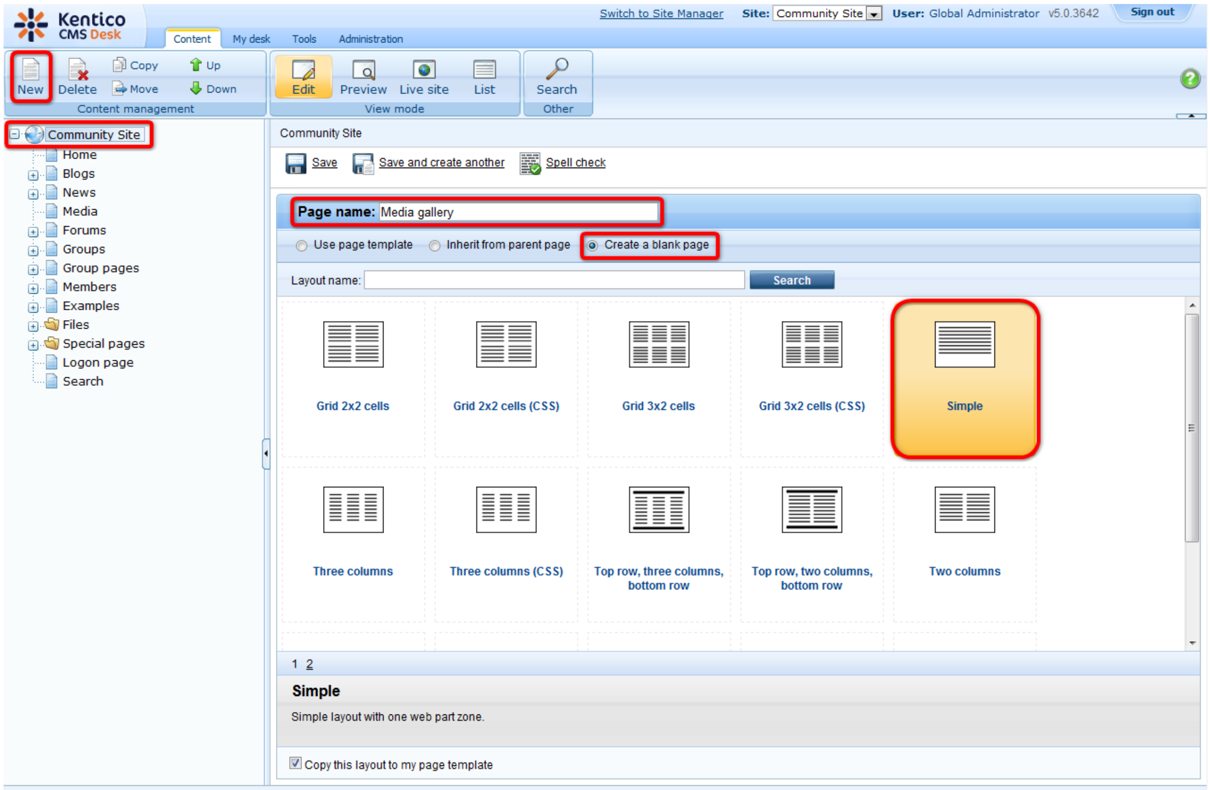
- **Thumbnail**: __thumbnails/MyImage_jpg_preview.bmp
- **Resized thumbnail**: __thumbnails/MyImage_jpg_preview_20_30.bmp

8.25.4 Displaying files from media library

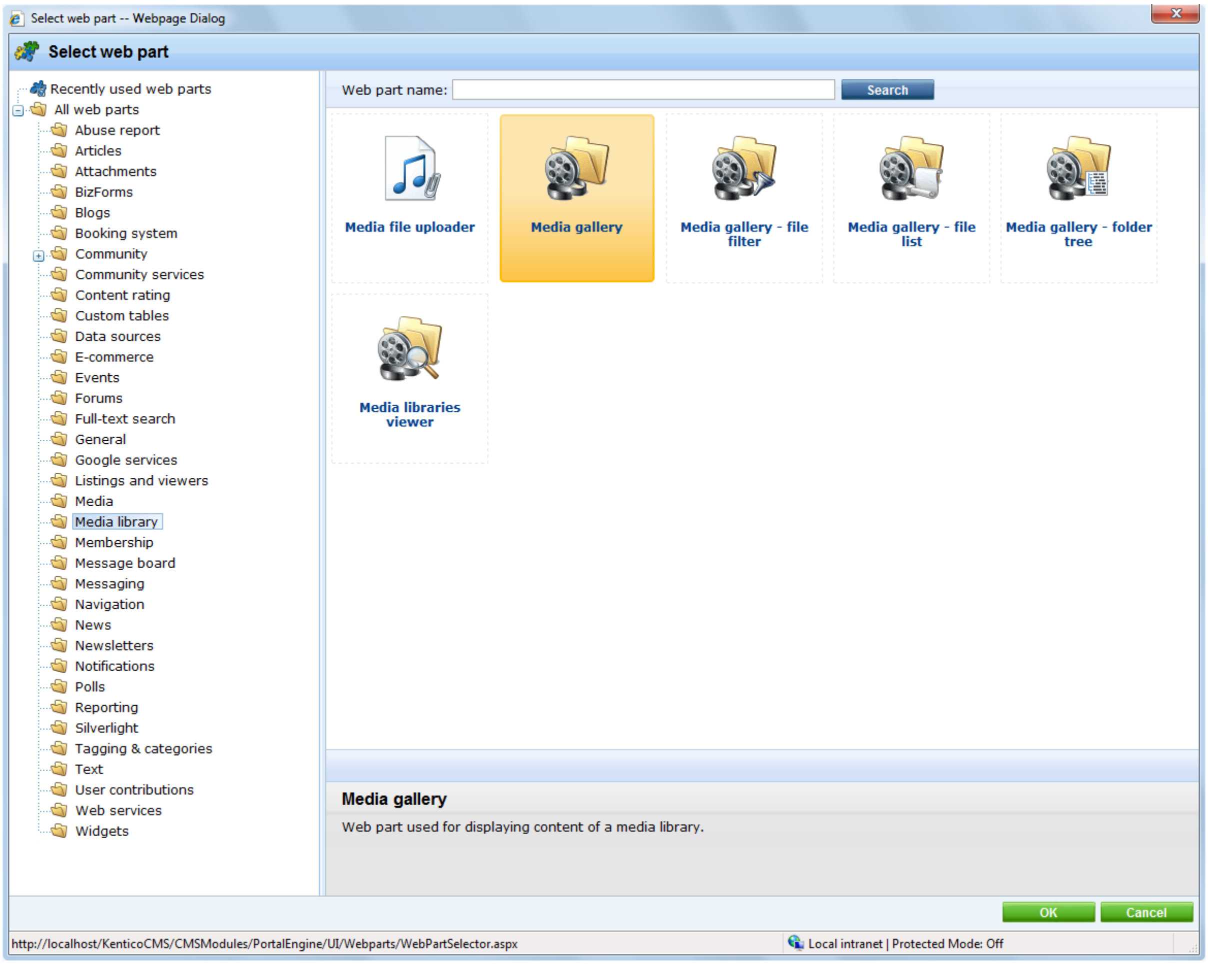
8.25.4.1 Using Media library web part

The **Media gallery** web part can be used to display content of media libraries on the live site. In the following example, you will learn how to let site visitors view content of a media library using this web part. We will use the sample **Community Starter site** as the base for our example.

1. Go to **CMS Desk**. Create a new **Page (menu item)** document in the root of the content tree and name it *Media gallery*. Choose to **Create a blank page** and use the **Simple** layout.



2. Click the **Add web part** () icon at the top right corner of the only web part zone on the page and select **Media library -> Media gallery**. Click **OK**.

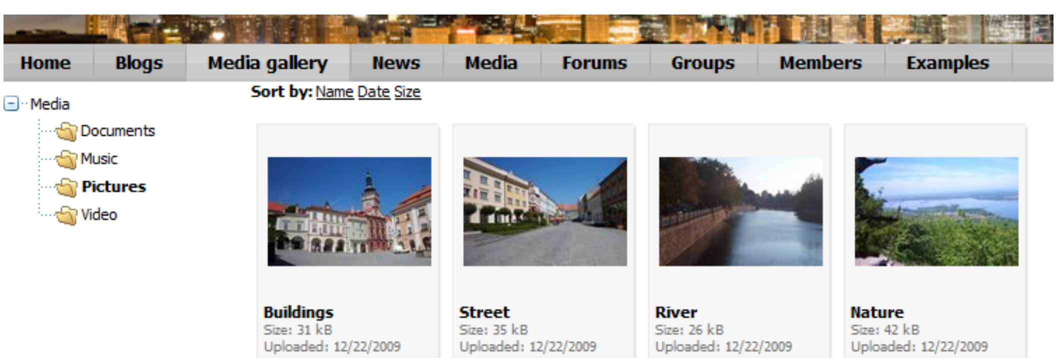


3. In the web part properties window, you only have to choose the media library that you want to display:

- **Media library** - Media

Click **OK**.

4. And that's it! If you switch to the live site, and select **Pictures** for instance, you should see the web part displaying content of the selected media library.



8.25.4.2 Using WYSIWYG editor

Media library files can be inserted into text using the WYSIWYG editor's **Insert image or media** and **Insert link** dialogs. You can find more information in the WYSIWYG editor chapter.

8.25.5 Settings

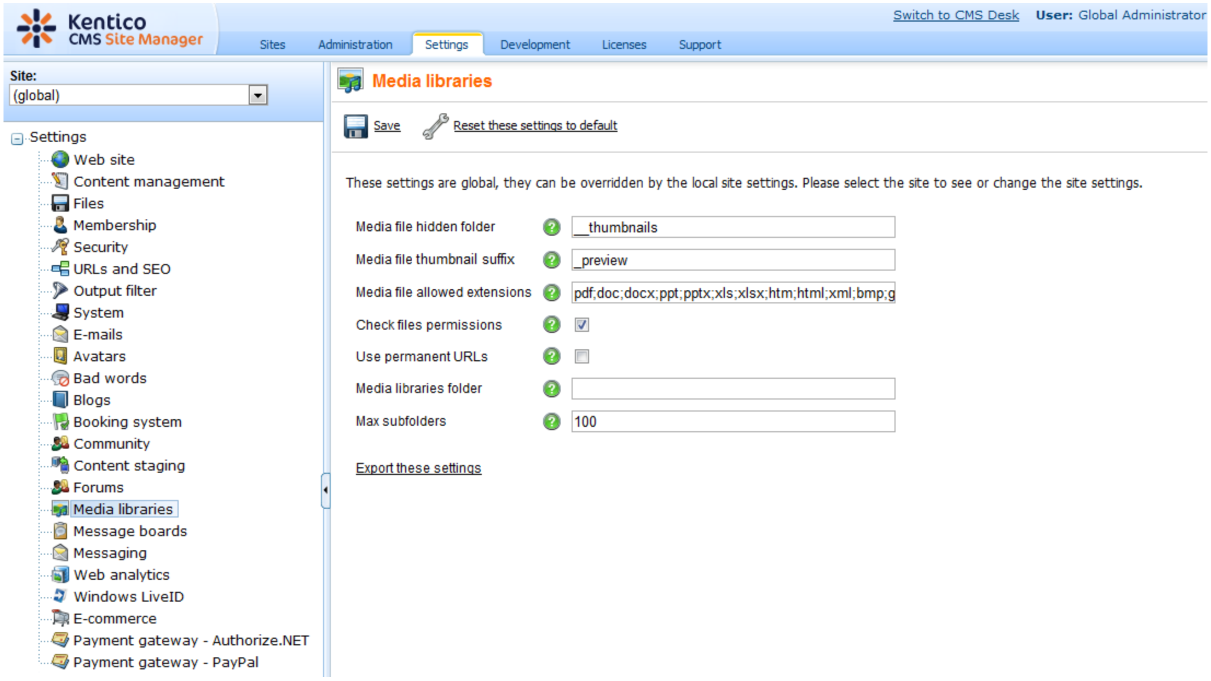
8.25.5.1 Media library settings

Settings related to the Media libraries module are located in **Site Manager -> Settings -> Media libraries**. The following settings are related to the module:

Media file hidden folder	Name of the folder where thumbnails of media files will be stored; this folder is hidden in the file system by default and thumbnails are generated within it.
Media file thumbnail suffix	Suffix added to thumbnail files; thumbnail files' names are in the following format: *<file name>_<file extension><thumbnail suffix>.<thumbnail extension>*
Media file allowed extensions	Extensions of files which can be uploaded to media libraries; should be entered divided by semicolons.
Check files permissions	Indicates if the "See media library content" permission is checked when retrieving media files using permanent URLs.
Use permanent URLs	If true, URLs of medial library files will be generated in permanent format (*~/getmedia/<file guid>/<file name>.<file extension>.<files friendly URL extension>*), otherwise direct path to the file will be used (e.g.: *~/MySite/Media/MyLibrary/MyImage.jpg*).
Media libraries folder	Folder where media library files are stored. You can use: • **physical disk path** - e.g. c:\myfiles\mysite • **virtual path** - ~/UploadedFiles • **UNC path** - \\server\folder If no value is entered, the default location is: **~/<site code name>/media**
Max subfolders	Maximal number of folders that can be displayed under an expanded folder node in the tree view.

Please note

If the **Media libraries** folder settings are inherited from global settings, the whole folder is deleted after deletion of the site. This may result in deletion of various files from other sites.



8.25.5.2 Configuring custom storage for media library

The default location of all libraries of a particular site is **~/<sitename>/media**. You can customize the location by entering a path in **Site Manager -> Settings -> Media libraries -> Media libraries folder**.

You can enter the folder location in the following formats:

- **physical path**: e.g. *c:\Libraries*
- **virtual path**: e.g. *~/Libraries*
- **UNC path**: e.g. *\\<server name>\Libraries*

Please note

In case that you are running the system on a **web farm** and have **the same UNC root defined** on all servers, it is necessary to add the following key into your *web.config* in order for the files stored in the libraries not to be transferred when synchronizing the web farm content:

```
<add key="CMSWebFarmSynchronizeMediaFiles" value="false"/>
```

8.25.5.3 Configuring custom file types

You can allow custom media types by a simple modification to the site's **web.config** and **MediaControl.ascx** inline controls. Any media types can be enabled to be recognized by the system in case that you have the right player for the file type.

The following example shows how to enable .flv videos in the system.

1. Add the following keys into the **appSettings** section of your **web.config** file. Add your custom extensions for the particular document types here for the files to be recognized as image/audio/video by the system, as you did with the highlighted **flv** extension.

```
<add key="CMSImageExtensions" value="bmp;gif;ico;png;wmf;jpg;jpeg;tiff;tif" />
<add key="CMSAudioExtensions" value="wav;wma;mp2;mp3;mid;midi;mpga" />
<add key="CMSVideoExtensions" value="avi;mp4;mpg;mpeg;wmv;qt;mov;rm;flv" />
```

2. Add the **flv** extension to the enumeration in **Site Manager -> Settings -> Media libraries -> Media file allowed extensions**, which will allow upload of these files into media libraries, and click the **Save** button.

You may also want to add the extension to the **Site Manager -> Settings -> Files -> Upload extensions** enumeration, which will allow upload of these files as CMS.File documents and document attachments, and click the **Save** button.

3. Download a player for your media type. In this example, you will use the JW FLV Player available here: http://www.longtailvideo.com/players/jw-flv-player/.

4. Extract the **swfobject.js** and **player.swf** files and put them in a folder under your website folder, e.g. **~/FlvPlayer**. This path will be used in the **CreateFlvObject()**, which you will create in step 6.

5. Open the **~/CMSInlineControls/MediaControl.ascx.cs** inline control in Visual Studio and replace the **Page_PreRender** method with the following code. It is the original code of the method with the first condition added. The added code ensures that when the file extension is **.flv**, the **CreateFlvObject()** method is called.

[C#]

```
protected void Page_PreRender(object sender, EventArgs e)
{
    if ((this.Type != null) && (this.Type.TrimStart('.').ToLower() == "flv"))
    {
        CreateFlvObject();
    }
    else if (MediaHelper.IsFlash(this.Type))
    {
        CreateFlash();
    }
    else if (ImageHelper.IsImage(this.Type))
    {
        CreateImage();
    }
    else
    {
        CreateMedia();
    }
}
```

6. Now you need to add the **CreateFlvObject()** private method to the control. In this example, it looks like the code sample below.

The method creates a new DIV tag for the FLV player, with ID generated in the divID variable. It also handles the player URL in a special way - the URL must be absolute and must end with *flv*. Support for the Autoplay and Loop properties of the player is also ensured.

[C#]

```
private void CreateFlvObject()
{
    string playerID = Guid.NewGuid().ToString("N");
    string flvUrl = UrlHelper.GetAbsoluteUrl(this.Url)
                .Replace("?", "%3F")
                .Replace("=", "%3D")
                .Replace("&", "%26");

    string flashVars = "file=" + flvUrl + "&provider=video&";
    flashVars += "controlbar=" + (AVControls ? "bottom" : "none") + "&";
    flashVars += "autostart=" + (AutoPlay ? "true" : "false") + "&";
    flashVars += "repeat=" + (Loop ? "always" : "none");

    string player = "";
    player += "<object id=\"" + playerID + "\"";
    player += "classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\"";
    player += "codebase=\"http://download.macromedia.com/pub/";
    player += "shockwave/cabs/flash/swflash.cab#version=9.0.115\"";
    player += "width=\"" + Width + "\" height=\"" + Height + "\">";
    player += "<param name=bgcolor value=\"#FFFFFF\">";
    player += "<param name=movie value=\"" + ResolveUrl("~/FlvPlayer/player.swf")
+ "\">";
    player += "<param name=allowfullscreen value=\"true\">";
    player += "<param name=allowscriptaccess value=\"always\">";
    player += "<param name=\"flashvars\" value=\"" + flashVars + "\">";
    player += "<embed name=\"" + playerID + "\" ";
    player += "type=\"application/x-shockwave-flash\" ";
    player += "pluginspage=\"http://www.macromedia.com/go/getflashplayer\" ";
    player += "width=\"" + Width + "\" height=\"" + Height + "\" ";
    player += "bgcolor=\"#FFFFFF\" ";
    player += "src=\"" + ResolveUrl("~/FlvPlayer/player.swf") + "\" ";
    player += "allowfullscreen=\"true\" ";
    player += "allowscriptaccess=\"always\" ";
    player += "flashvars=\"" + flashVars + "\">";
    player += "</embed>";
    player += "</object>";

    this.ltlMedia.Text = player;
}
```

7. To make this work also on the live site using media libraries transformations, you now need to open the **~/CMSModules/MediaLibrary/Controls/LiveControls/MediaFilePreview.ascx.cs** file and replace the following line of code:

```
else if (CMS.GlobalHelper.MediaHelper.IsVideo(mfi.FileExtension))
```

with this piece of code:

```
else if (mfi.FileExtension.TrimStart('.').ToLower() == "flv") {
    this.ltlOutput.Text = CreateFlvObject(url);
}
else if (CMS.GlobalHelper.MediaHelper.IsVideo(mfi.FileExtension))
```

8. Similarly, you will need to add the **CreateFlvObject(string url)** private method to the control. In this example, it looks like the code sample below.

The method creates a new DIV tag for the FLV player, with ID generated in the divID variable. It also handles the player URL in a special way - the URL must be absolute and must end with *flv*. Please note that the Autoplay and Loop properties of the player are defined firmly in the code.

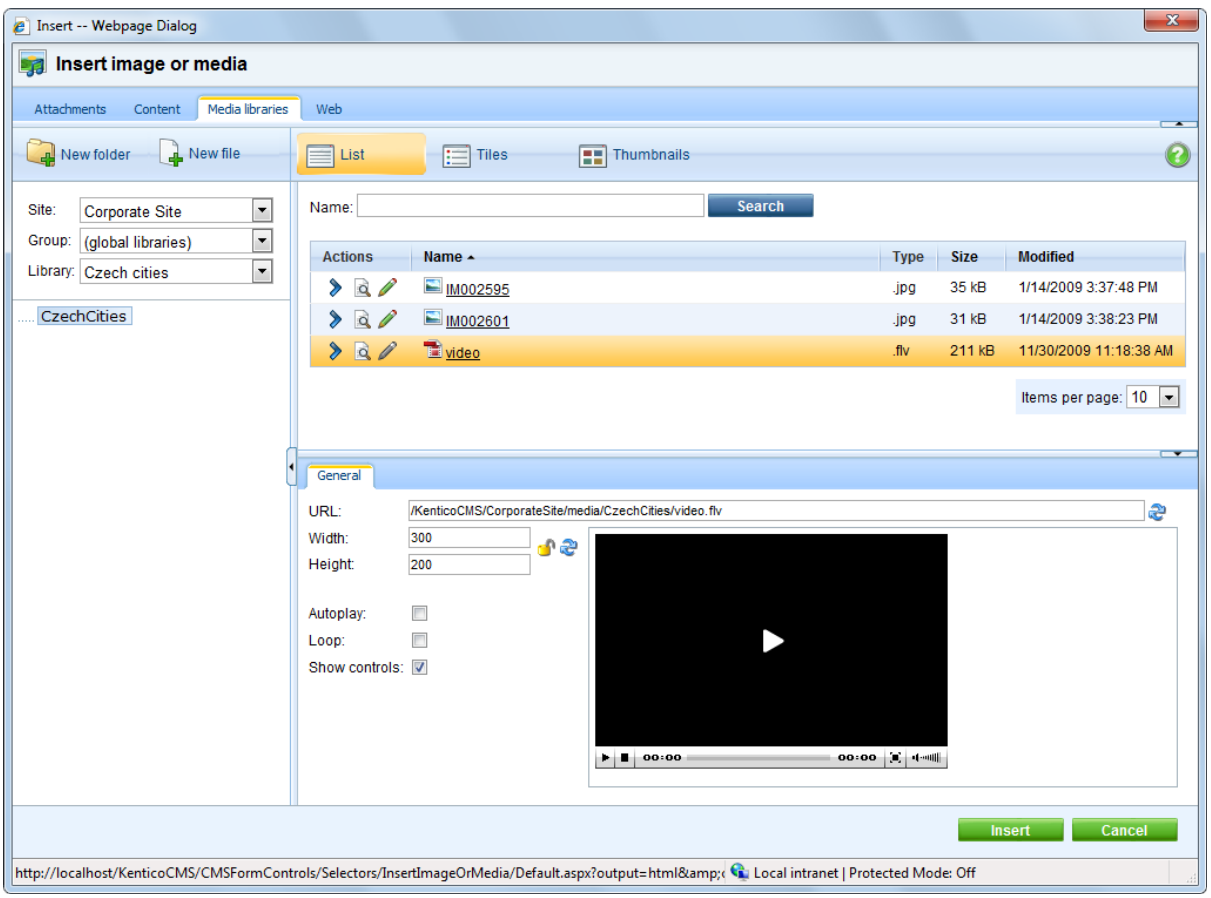
```
private string CreateFlvObject(string url)
{
    string playerID = Guid.NewGuid().ToString("N");
    string flvUrl = UrlHelper.GetAbsoluteUrl(url)
                .Replace("?", "%3F")
                .Replace("=", "%3D")
                .Replace("&", "%26");

    bool avControls = true;
    bool autoPlay = false;
    bool loop = false;

    string flashVars = "file=" + flvUrl + "&provider=video&";
    flashVars += "controlbar=" + (avControls ? "bottom" : "none") + "&";
    flashVars += "autostart=" + (autoPlay ? "true" : "false") + "&";
    flashVars += "repeat=" + (loop ? "always" : "none");

    string player = "";
    player += "<object id=\"" + playerID + "\"";
    player += "classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\"";
    player += "codebase=\"http://download.macromedia.com/pub/";
    player += "shockwave/cabs/flash/swflash.cab#version=9.0.115\"";
    player += "width=\"" + Width + "\" height=\"" + Height + "\">";
    player += "<param name=bgcolor value=\"#FFFFFF\">";
    player += "<param name=movie value=\"" + ResolveUrl("~/FlvPlayer/player.swf")
+ "\">";
    player += "<param name=allowfullscreen value=\"true\">";
    player += "<param name=allowscriptaccess value=\"always\">";
    player += "<param name=\"flashvars\" value=\"" + flashVars + "\">";
    player += "<embed name=\"" + playerID + "\" ";
    player += "type=\"application/x-shockwave-flash\" ";
    player += "pluginspage=\"http://www.macromedia.com/go/getflashplayer\" ";
    player += "width=\"" + Width + "\" height=\"" + Height + "\" ";
    player += "bgcolor=\"#FFFFFF\" ";
```

```
    player += "src=\"" + ResolveUrl("~/FlvPlayer/player.swf") + "\" ";
    player += "allowfullscreen=\"true\" ";
    player += "allowscriptaccess=\"always\" ";
    player += "flashvars=\"" + flashVars + "\">";
    player += "</embed>";
    player += "</object>";

    return player;
}
```

If you need to change the behavior of the player, please modify these lines of code of the added method.

```
bool avControls = true;
bool autoPlay = false;
bool loop = false;
```

9. Now .flv files are recognized as videos by the system. The custom player is used for the files on the live site and also in the **Insert image or media** dialog, as you can see in the screenshot below.



8.25.5.4 Configuring maximal uploaded file size

The default maximal uploaded file size setting on IIS6/7 is 30MB. To enable uploads of larger files, which is essential for the functionality that the media libraries were designed for, you need to add the following keys to your **web.config** file:

IIS 6

All you need to do is to set the value of the following property, which is already present in your **web.config**, to the desired value, while the value is entered in **kiloBytes**:

```
<httpRuntime maxRequestLength="2000000" />
```

IIS 7

You need to do the same setting as described above for IIS 6. Besides, you also need to add the following highlighted code to the end of your **web.config**. The value of the **maxAllowedContentLength** property is entered in **Bytes**:

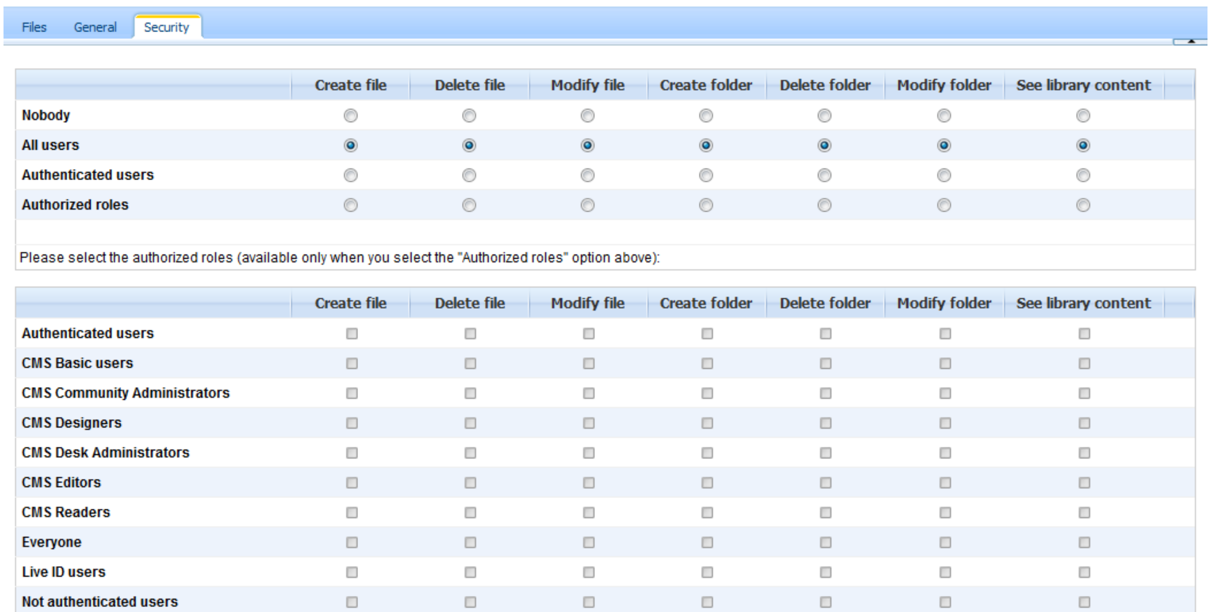
```
...

<system.webServer>
<security>
    <requestFiltering>
      <requestLimits maxAllowedContentLength="2147483648"/>
    </requestFiltering>
</security>
  <validation validateIntegratedModeConfiguration="false" />
  <modules>
    <add name="ScriptModule" preCondition="integratedMode" type="System.Web.
Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
    <add name="XHtmlModule" type="CMS.CMSOutputFilter.OutputFilterModule, CMS.
OutputFilter" />
  </modules>
  <handlers>
    <remove name="WebServiceHandlerFactory-Integrated" />
    <add name="ScriptHandlerFactory" verb="*" path="*.asmx" preCondition="
integratedMode" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.
Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
    <add name="ScriptHandlerFactoryAppServices" verb="*" path="*_AppService.axd"
preCondition="integratedMode" type="System.Web.Script.Services.
ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
    <add name="ScriptResource" preCondition="integratedMode" verb="GET,HEAD" path=
"ScriptResource.axd" type="System.Web.Handlers.ScriptResourceHandler, System.Web.
Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
  </handlers>
</system.webServer>
</configuration>
```

8.25.6 Security

8.25.6.1 Media library permissions

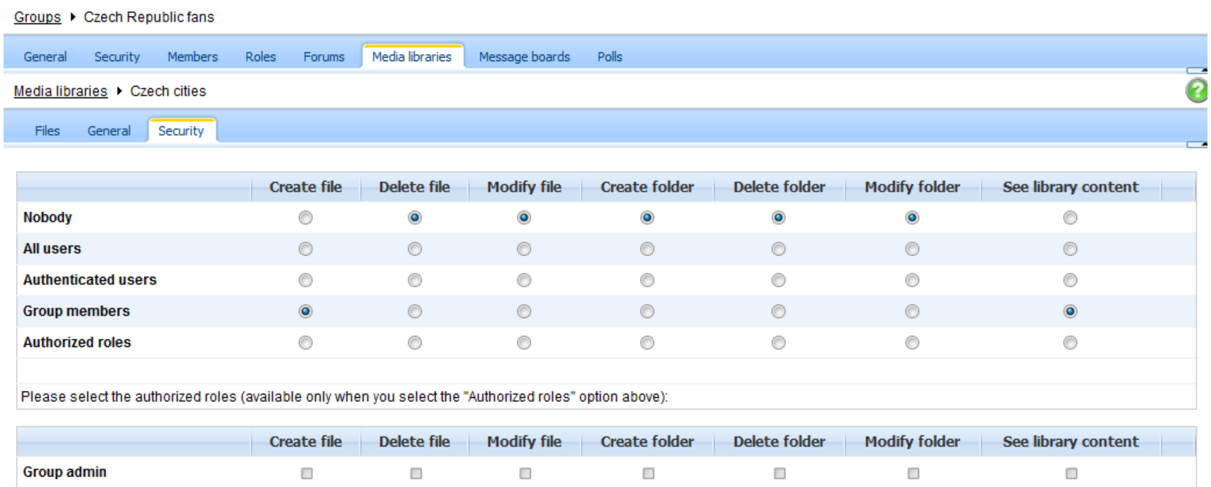
On the Security tab of each media library, you can assign permissions for particular actions to:

- **Nobody** - nobody can perform the action
- **All users** - all users including anonymous site visitors can perform the action
- **Authenticated users** - only signed-in site members can perform the action
- **Authorized roles** - only members of roles selected by the check-boxes below can perform the action



When editing media library of some group, permissions for particular actions can be assigned to:

- **Nobody** - nobody can perform the action
- **All users** - all users including anonymous site visitors can perform the action
- **Authenticated users** - only signed-in site members can perform the action
- **Group members** - only members of the group can perform the action
- **Authorized roles** - only members of group roles selected by the check-boxes below can perform the action



The following table shows which permissions need to be assigned to allow users to perform particular actions. Global administrators can perform all of these actions for all general and group media libraries on the site. Group administrators can perform all of these actions for group media libraries of groups where they are group administrators.

Action/ Permission				File			Folder			
	Read	Manage		Create	Delete	Modify	Create	Delete	Modify	See library content
Media library content administration - files										
upload		X	or	X						
rename / change title and description		X	or			X				
delete		X	or		X					
copy		X	or	X						
move		X	or			X				
Media library content administration - folders										
create		X	or				X			
rename		X	or						X	
delete		X	or					X		

copy		X	or				X			
move		X	or						X	
CMSDesk administration										
See media libraries UI	X									
Modify media libraries UI		X								
Live site administration										
See media libraries UI	X									
Modify media libraries UI		X								
CMSDesk & Live site										
Select image/flash/ media dialogs in WYSIWYG editor	X									
Live site										
See and browse library content (Media gallery webpart)	X		or							X
Upload file (Media file uploader webpart)		X	or	X						

Module permissions

The following permissions can be set in **Site Manager -> Administration -> Permissions,** when you choose the **Modules -> Media libraries** permission matrix:

- **Manage** - allows creating, deleting and editing media libraries and managing media library content via the administration interface
- **Read** - allows seeing media library content and its properties when editing a media library, but does not allow to make any changes to it

	Manage	Read
Authenticated users		
CMS Basic users		
CMS Community Administrators	✓	✓
CMS Designers		
CMS Desk Administrators		✓
CMS Editors		✓
CMS Readers		✓
Everyone		
Gold Partners		
Live ID users		
Not authenticated users		
Silver Partners		

8.25.6.2 Secured vs. Non-secured libraries

Media libraries on your site can be secured or non-secured. To ensure the required functionality, several settings need to be done as described below.

Please note!

By default, files in media libraries are NOT secured and can be accessed directly by anybody who knows the exact link to the file. If you want to prevent this behavior, please set up your media library as a secured one.

Secured libraries

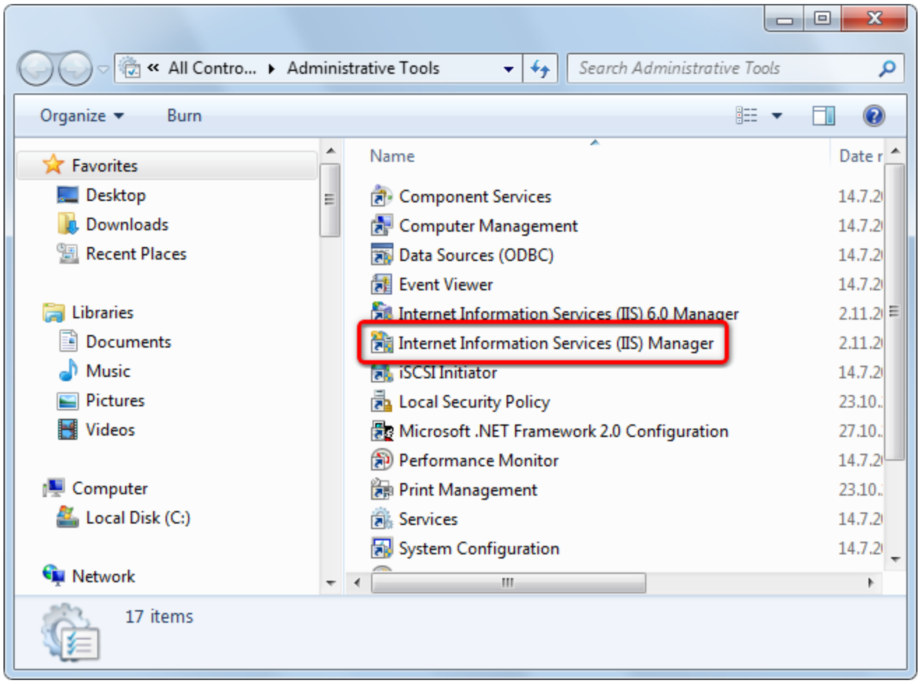
Secured media libraries allow viewing of their content only to members of authorized roles or only to authenticated users, based on the settings made on the library's **Security** tab. Secured libraries are also slower than the non-secured ones as permission checking involves some processing overhead.

To set up a media library to behave as a secured library, you have to take the following steps:

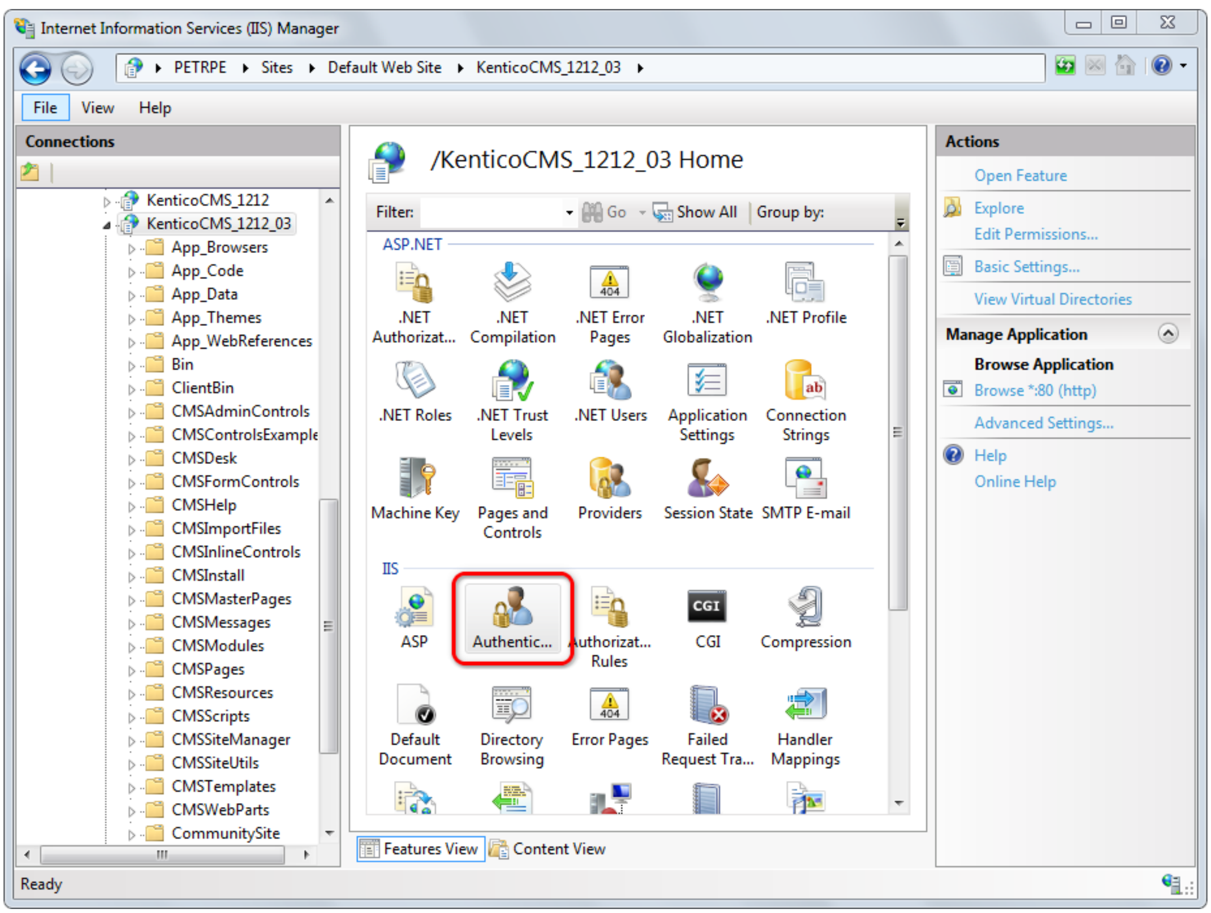
1. IIS setup

You have to **set up your IIS** so that files can not be downloaded directly from the library by typing the link to the file like <*site url*>*/media/file.jpg* into the browser.

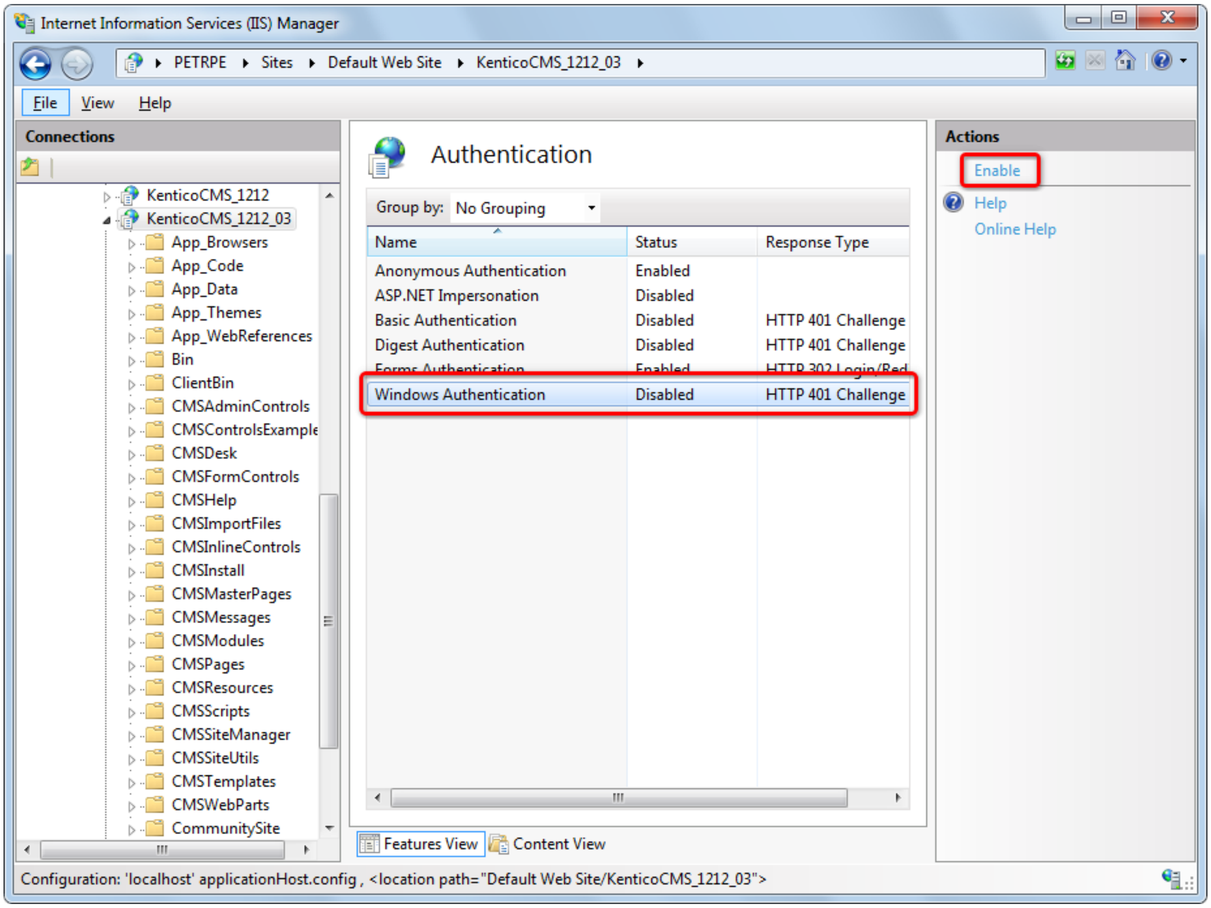
a) Go to **Start** -> **Control Panel** -> **Administrative Tools** and start the **Internet Information Services (IIS) Manager**.



b) Locate and select the media library folder in the IIS tree, then click on the **Authentication** icon.

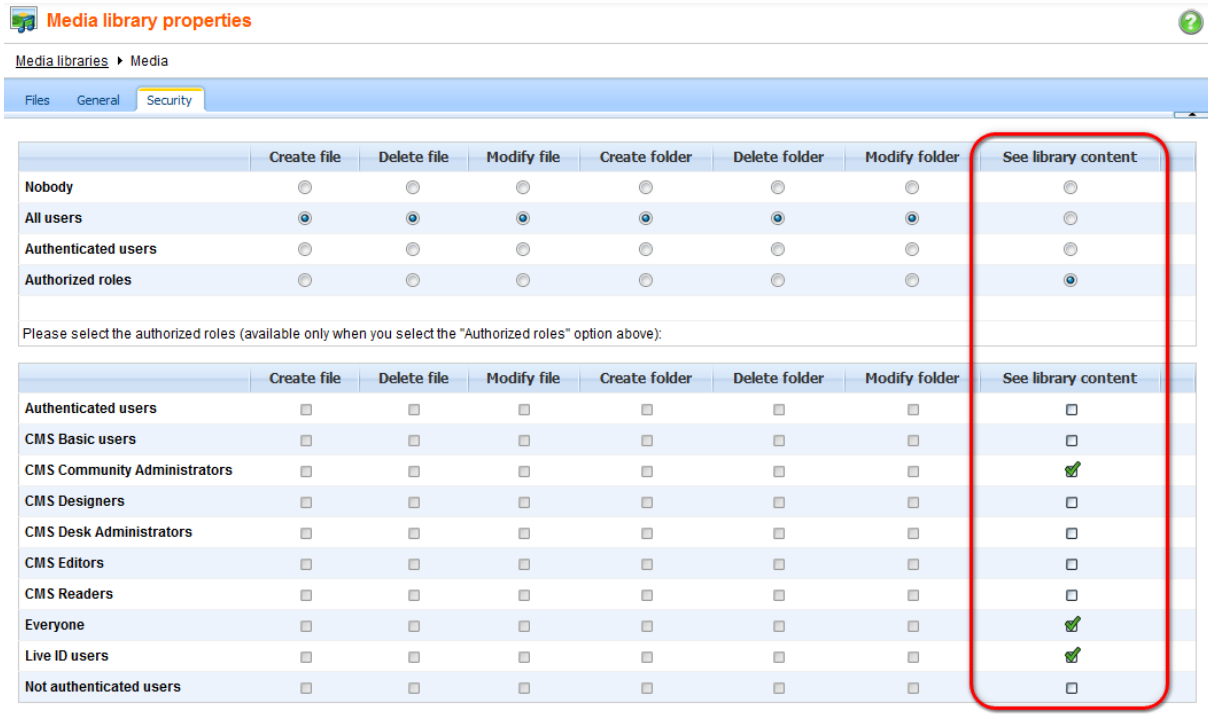


c) Disable the **Anonymous Authentication** by clicking the **Disable** link in the **Actions** list. If enabled, disable also the **Windows Authentication**.



2) Media library security settings

You have to assign the **See library content** permission to the appropriate roles or all authenticated users on the media library's **Security** tab.



You also have to make sure that the **Check files permissions** option is enabled in **Site Manager -> Settings -> Media libraries**. With this option disabled, permission checks would not be performed.

3) Media gallery web part settings

a) You need to enable the **Use secure links** web part property.

b) When writing your transformations for the **Media gallery** web part, you should stick to the following rules:

File previews and **file details** should be displayed using the following control:

```
<cc1:MediaFilePreview ID="filePreview" runat="server" maxsidesize="117" />
```

Download links should be obtained using the following method:

```
<%# MediaLibraryFunctions.GetMediaFileUrl(Eval("FileLibraryID") ,Eval("FilePath"),
Eval("FileGUID"), Eval("FileName"), GetDataControlValue<bool>("UseSecureLinks")) %
>
```

You can see an example of a real-world use of this web part, including the defined transformations, on the **Community Site** sample website, in the **Media** section.

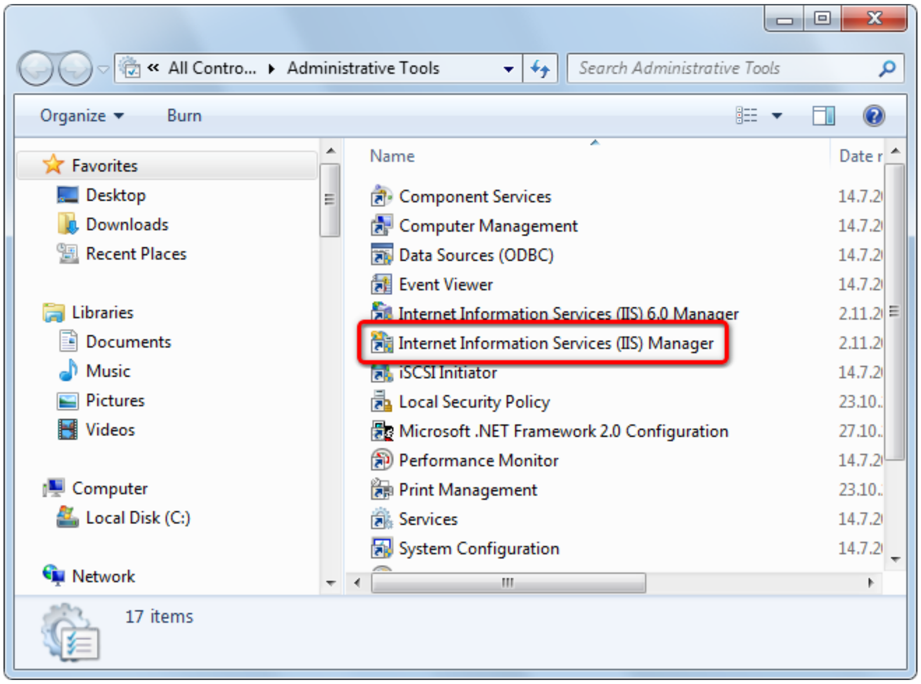
Non-secured libraries

Content of non-secured media libraries is accessible to all site users or visitors. These libraries are also faster than the secured ones, as no permissions need to be checked.

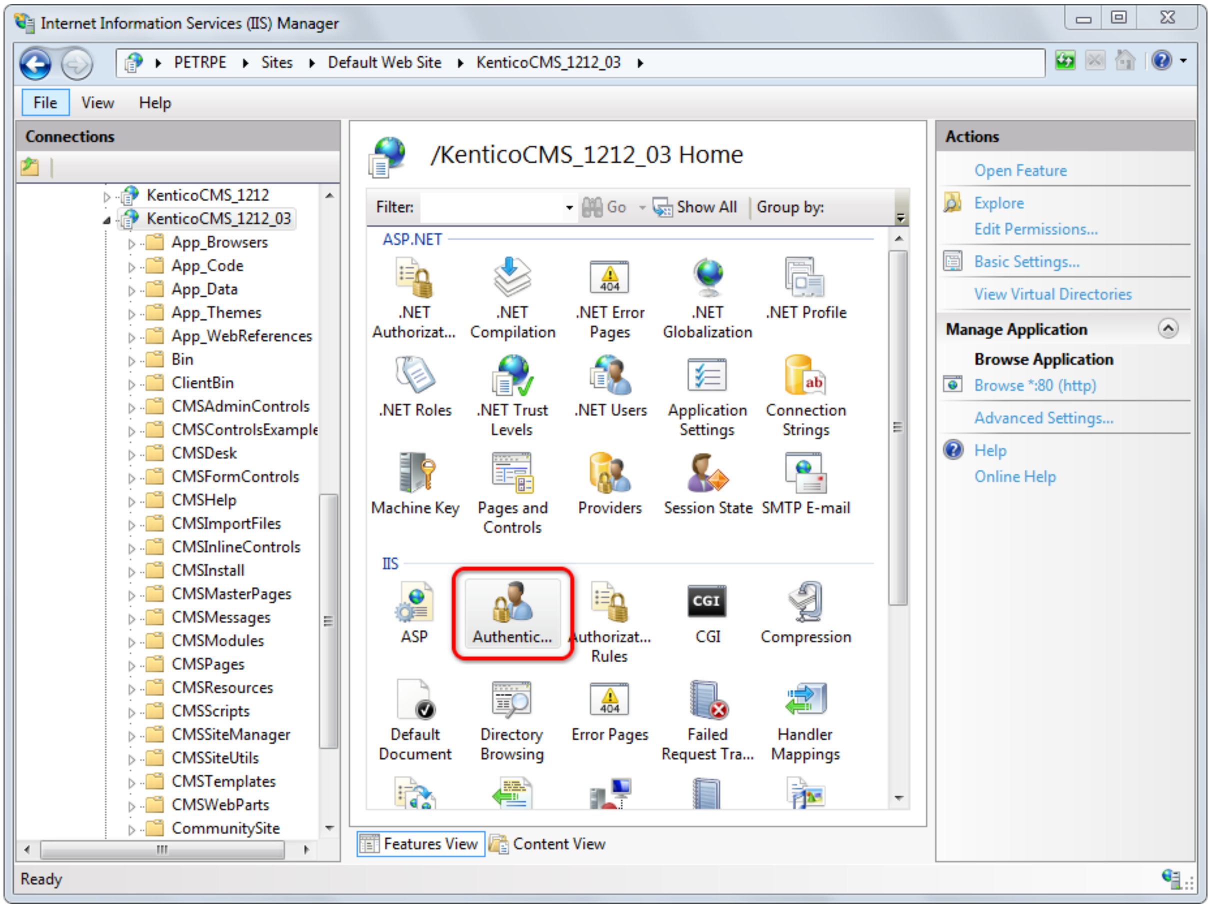
1. IIS setup

You have to **set up your IIS** so that files in the library can be accessed directly by anonymous users.

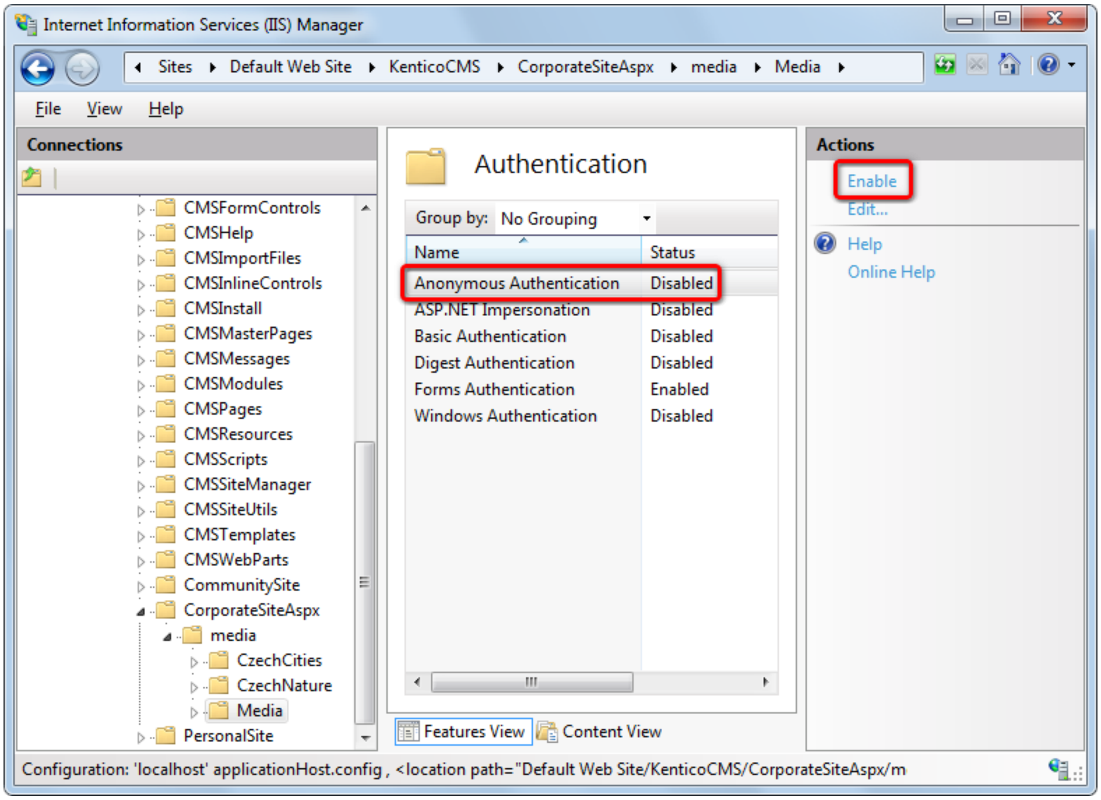
a) Go to **Start** -> **Control Panel** -> **Administrative Tools** and start the **Internet Information Services (IIS) Manager**.



b) Locate and select the media library folder in the IIS tree, then click on the **Authentication** icon.

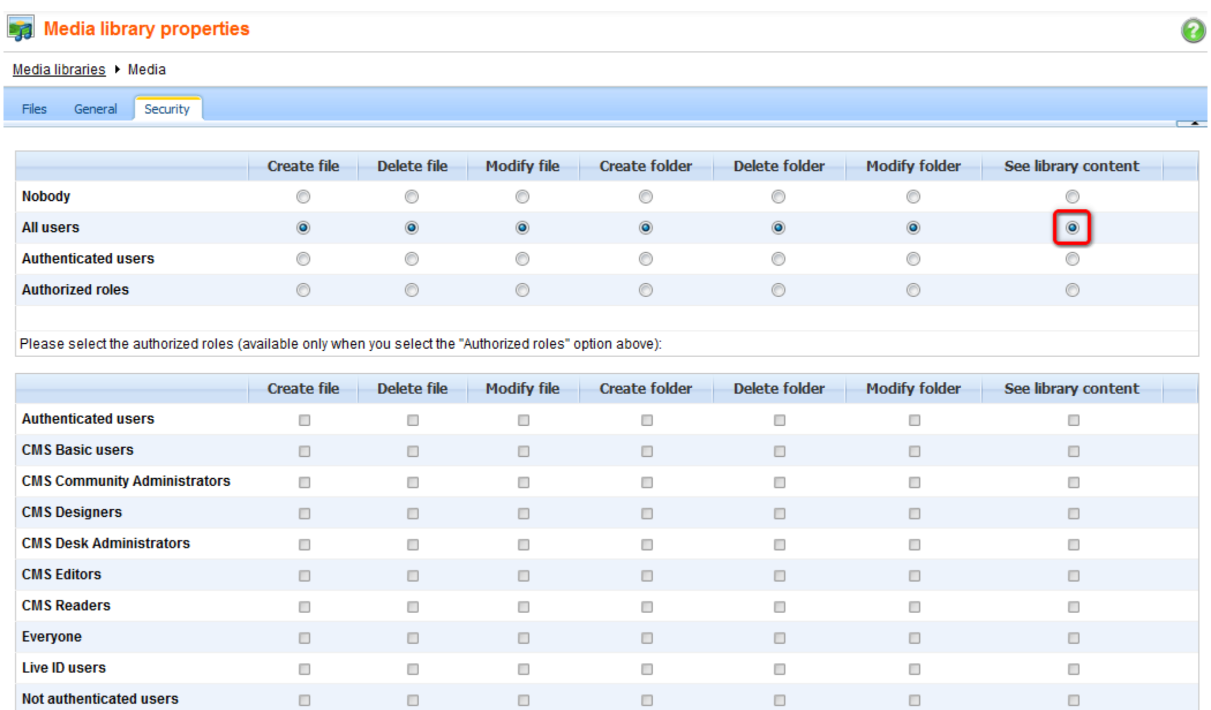


c) Enable the **Anonymous Authentication** by clicking the **Enable** link in the **Actions** list.

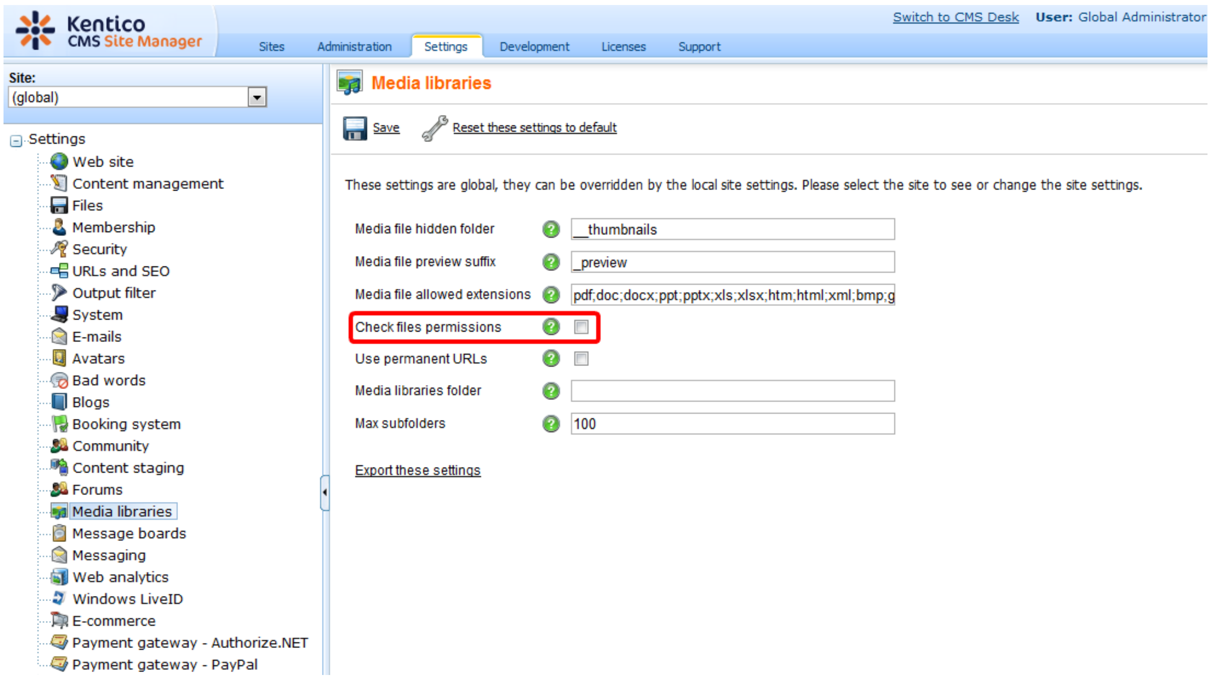


2) Media library security settings

You have to assign the **See library content** permission to **All users** on the media library's **Security** tab.



Alternatively, if all media libraries on the site are non-secured, you can disable the **Check files permissions** option in **Site Manager -> Settings -> Media libraries**. This disables all permission checks for all media libraries on the site, which enables all users to see the libraries' content.



3) Media gallery web part settings

a) The **Use secured links** web part property should be disabled for the file requesting to be faster.

b) When writing your transformations for the **Media gallery** web part, you should stick to the following rules:

Image previews and **image details** should be obtained using the following method because they need to be resized:

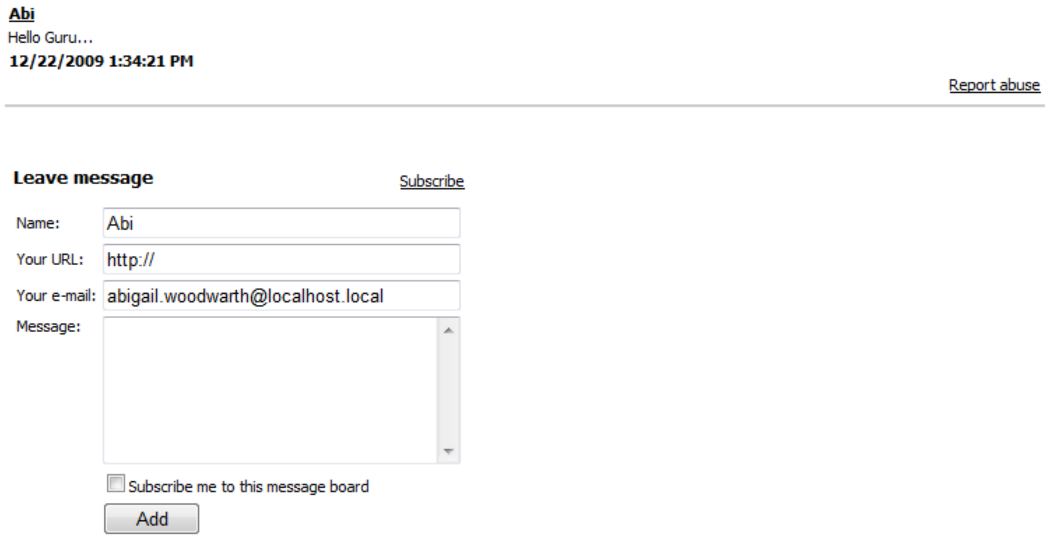
```
<cc1:MediaFilePreview ID="filePreview" runat="server" maxsidesize="117" />
```

Other file types' previews and **details** can be obtained using direct links. **Download links** can be obtained directly too.

8.26 Message boards

8.26.1 Overview

The Message boards module enables site visitors to comment on the content of a particular page. It means you can insert comments on specific document content, on a specific user or a group of users etc.



Among web parts that come with the Message boards module are the following web parts:

- **Message board -** displays messages on the board and allows adding new messages; used for general documents or for user profiles
- **Group message board -** similar to Message board, but optimized for group profiles
- **Message board viewer -** displays messages from all message boards on the specified site, but doesn't allow adding new messages
- **Group message board viewer -** similar to Message board viewer, but optimized for group profiles
- **Message board unsubscription -** displays a confirmation that the user has been successfully unsubscribed from the message board

The Message boards module provides similar functionality like the Forums module. However unlike structured forums, message boards are flat and more straightforward. They are created automatically after the first comment is inserted or on subscribing without leaving a comment.

Each message board is related to a document on which it is placed, and optionally to a user or a group. Board messages can be moderated by board moderators. Users can also subscribe to receiving notifications about new messages and rate the content of the document the same way as when using the Content rating web part.

In the Message boards chapter, you can learn how to use the **Message board** web part (learn here how to do it), manage message boards (as explained here), edit message boards (as explained here) and how to set Board base URL (as explained here). Topics on the settings and security of the the Message boards module can be found in this chapter, too. Finally, the Message board notifications subchapter will guide you through the process of setting up the module to receive notifications about new messages added to the message board (for reference, see the following topics: Who can be notified, User subscriptions, E-mail templates).

If you would like to see Message boards module sample sites, please switch to the Corporate site sample website and in the content tree go to **Examples -> Web parts -> Message boards** to display the showcase of the Message boards module functionality. If you would like to see an example of managing a message board and its messages on Live site using the **Group profile** web part, please switch to the Community site sample website and choose **Groups -> <some group>** from the main menu.

Another example of the use of the Message boards module is available in Kentico CMS Community Site Guide; please note that these practical examples do not concern the whole functionality of the module but focus on its use in a broader context of the Community Site sample website:

- See Community Site Guide -> Part 2 -> Creating the Special pages -> Creating the Board unsubscribe page: An example of handling message board unsubscription requests.
- See Creating the Members section -> Creating the Profile page in the same section of the Community Site Guide: An example of adding and setting up two **Message board** web parts.
- See Creating the Groups section -> Creating the Profile page: An example of adding and setting up a **Group message board** web part.

8.26.2 Using the Message board web part

Message board can be placed on any page (document) on your website. It allows site visitors to send instant comments on the content of the particular page. You can have unlimited number of message boards on one page.

In the following example, you will see the basic process of adding a message board to your site. Our goal will be to add a Message board to the **Your first news** document on the sample **Corporate site**.

1. Log in to CMS Desk as *administrator* with blank password. From the content tree, select **News -> Your first news**.

2. Click the **Add web part** ( ) icon at the top right corner of **zoneLeft** web part zone and in the web part selection window, choose **Message board -> Message board** and click **OK**.

3. In the web part properties window, you can set a number of properties. As there is quite a lot of them, we will leave default values and set only those listed below. If you want exact explanation of the meanings of each of the properties, please refer to Kentico CMS Web Parts Reference or click the **Documentation** ( ) link at the top right corner of the web part properties window.

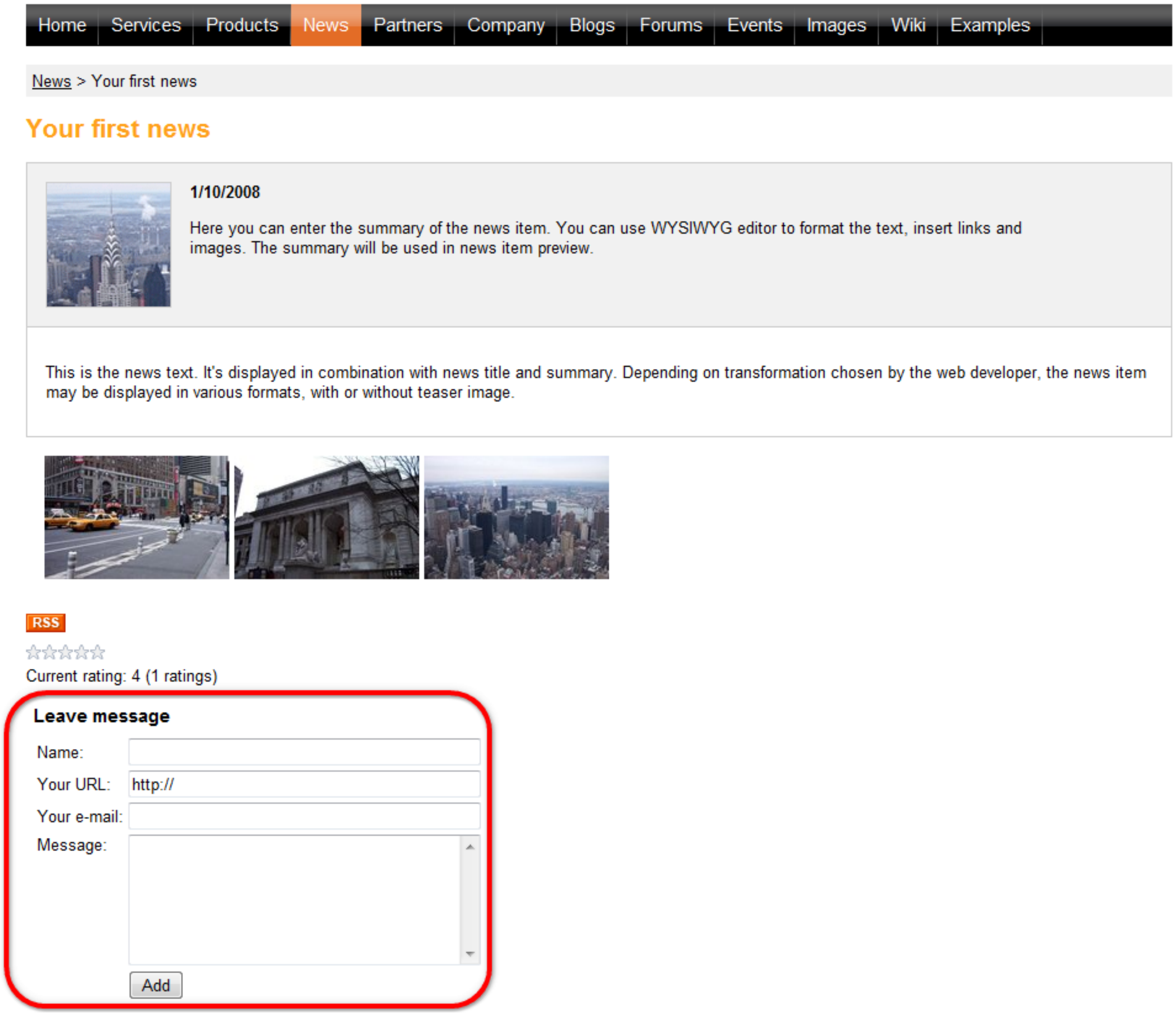
- **Message transformation** - transformation used for displaying board messages; you can use *Community.Transformations.MessageBoard*, which is a default predefined transformation for board messages
- **No messages text** - text message that will be displayed to site visitors when there are no messages in the board
- **Display name** - display name of the message board, it will be used in the administration interface to identify the message board, so it is advisable to use some well-descriptive name under that you will recognize the message board; it is useful to use macros, such as *{%SiteName%} - my first message board*

Click **OK** when you are finished entering the values.

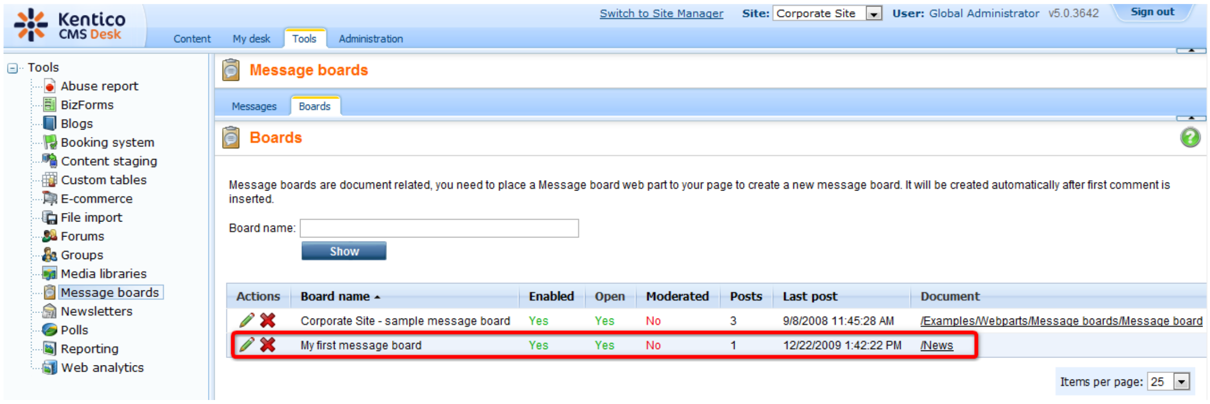
!! Board default settings !!

Properties in the **New board settings** section will be used when the message board is created (after first message or subscription). Thereafter, changes made to the web part properties will take no effect. To change some of these properties, you will have to edit the message board in **CMS Desk -> Tools -> Message boards -> Boards** tab.

4. If you switch to the **Live Site** mode, you should see the form for entering board messages and the 'No messages text' displayed above it, as you can see in the following screenshot.



5. If you go to the administration interface located in **CMS Desk -> Tools -> Message boards**, you won't see the message board yet as message boards are created in the administration interface section **after first message is added** to it, **not after adding the web part** to the site. Try adding some message now. After doing so, go to **CMS Desk -> Tools -> Message boards**. You should see the new message board in the list as in the following screenshot.



8.26.3 Managing message boards

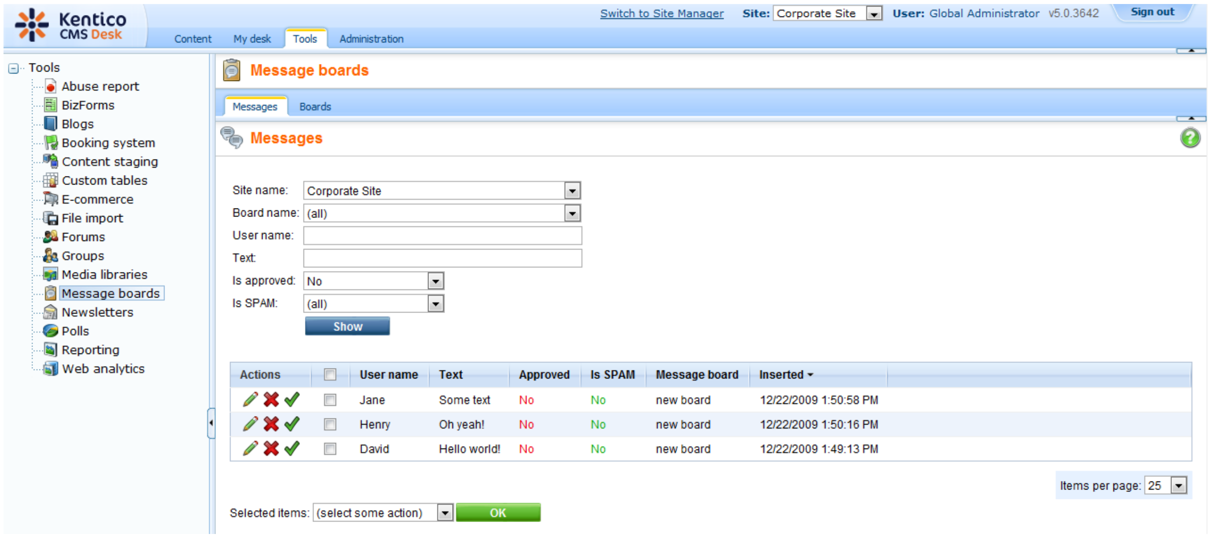
Message boards management can be performed in **CMS Desk -> Tools -> Message boards**. The section is divided into two tabs.

Messages tab

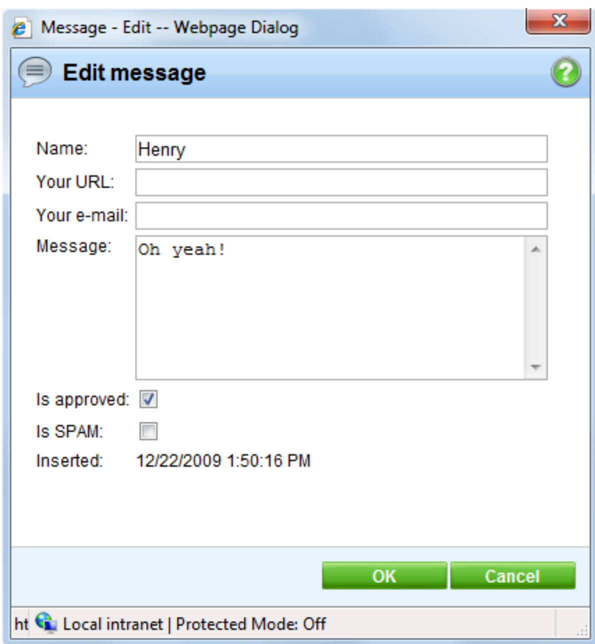
On this tab, moderators can manage board messages. By default, there are only messages requiring approval displayed when you access the page, so that the moderator sees only new messages that need to be approved or rejected. Until the messages are approved, they won't be displayed on the message board. Rejected messages won't be displayed either.

Using the filter above the list, you can determine which messages you want to display. The following filtering parameters are available:

- **Site name** - only messages from the selected site will be displayed
- **Board name** - only messages from the selected message board will be displayed
- **User name** - only messages posted by the user specified here will be displayed; you can also enter only a part of the user name
- **Text** - only messages containing the entered text will be displayed
- **Is approved** - you can choose whether to display only approved or disapproved messages
- **Is SPAM** - you can choose whether to display only messages marked or not marked as SPAM



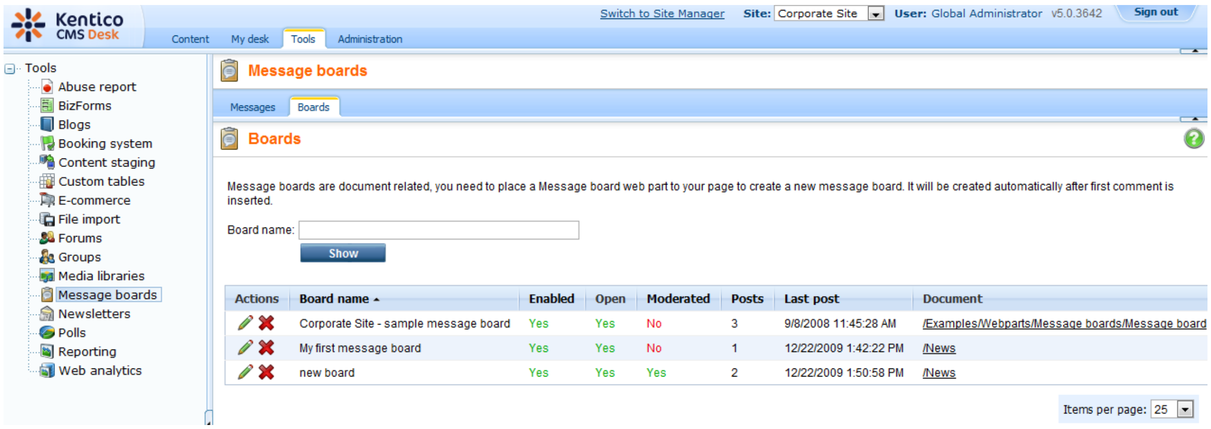
Messages in the list can be **Approved** (), **Rejected** (), **Deleted** () or **Edited** (). If you choose to edit a message, the following window pops up, letting you make changes to it. You can also select more messages using the check-boxes and perform one of these actions for all of them using the **Selected items** drop-down list and clicking the **OK** button.



Boards tab

On this tab, you can see a list of all message boards on the current site. Using the **Board name** field above the list, you can filter the displayed message boards. You don't need to enter the exact name, you can enter only a part of it and the list will display all message boards with name containing the entered expression. It is a good idea to give your message boards well-descriptive names so that you can tell one from another and search efficiently.

Message boards can be **Edited** () and **Deleted** () on this tab.



8.26.4 Editing message boards

You can edit message board properties at **CMS Desk -> Tools -> Message boards**, on the **Boards** tab. If you click the **Edit** ( ) icon next to some message board, its editing interface will be displayed.

The administration interface reflects the **Message board** web part properties. When you add the web part to a page, you can set its properties in the web part properties dialog. When the first message is added to the board, the message board is created in this section of the administration interface. At this point, changes made to the web part properties have no effect and you have to make all changes only in this section!

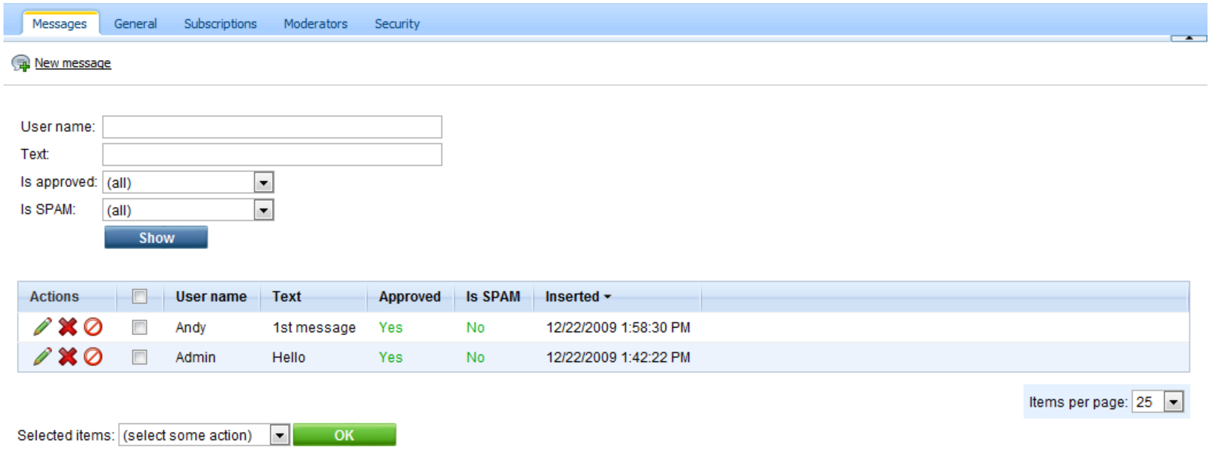
The administration interface for editing message boards is divided into five tabs:

Messages tab

This tab displays a list of all messages on the message board. Using the **New message** link, you can add new messages to the board directly from the administration interface. Below is a filter, letting you display only messages matching specified criteria. The following filtering parameters are available:

- **User name** - only messages posted by the user specified here will be displayed; you can also enter only a part of the user name
- **Text** - only messages containing the entered text will be displayed
- **Is approved** - you can choose whether to display only approved or disapproved messages
- **Is SPAM** - you can choose whether to display only messages marked or not marked as SPAM

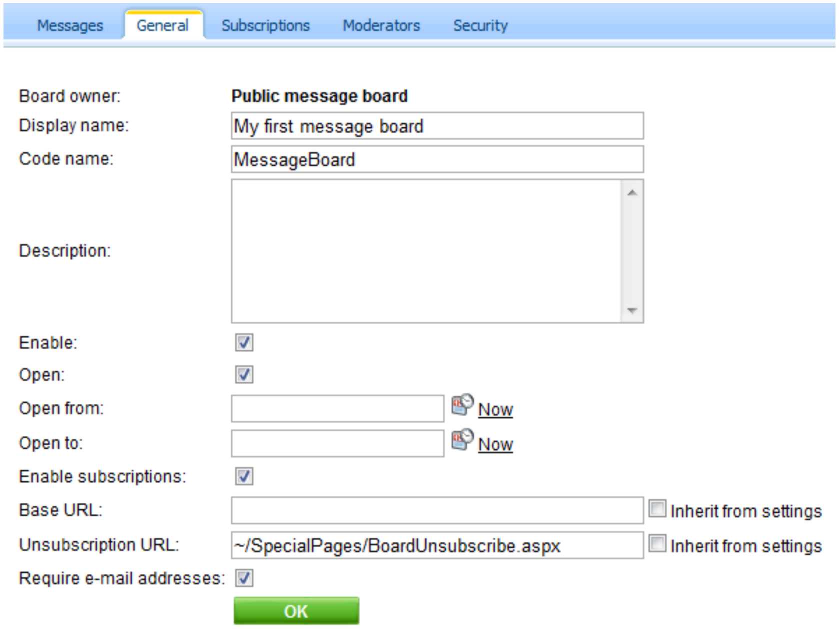
Messages can be **Edited** ( ), **Deleted** ( ), **Approved** ( ) or **Rejected** ( ). You can also select more messages using the check-boxes and perform one of these actions for all of them using the **Selected items** drop-down list and clicking the **OK** button.



General tab

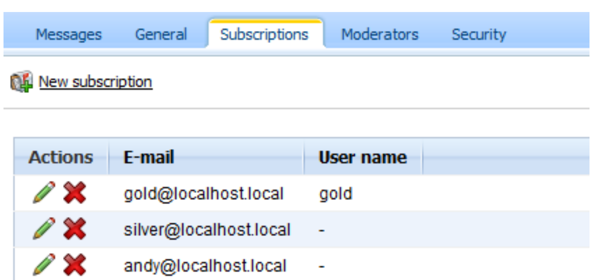
On the general tab, you can specify the following properties of the message board:

- **Display name** - display name of the message board
- **Code name** - code name of the message board
- **Description** - text describing the message board
- **Enable** - if unchecked, the message board will be hidden; if checked, the message board works normally
- **Open** - if checked, adding messages is enabled; if unchecked, messages are displayed but cannot be added
- **Open from / to** - using these fields, you can define the time interval during that new messages can be added to the board
- **Enable subscriptions** - if checked, users can subscribe to receiving notifications about new messages on the board
- **Base URL** - URL used as the URL base of links to message boards in notification e-mails; if empty, value from **Site Manager -> Settings -> Message boards -> Board base URL** will be used; if that property is empty too, message boards cannot be placed on pages with wildcard URLs
- **Unsubscription URL** - URL of the page containing the **Message board unsubscription** web part; the web part handles notification unsubscription requests; if not set, value in **Site Manager -> Settings -> Message boards -> Board unsubscription URL** will be used
- **Require e-mail addresses** - if checked, users are required to enter their e-mail address when posting board messages



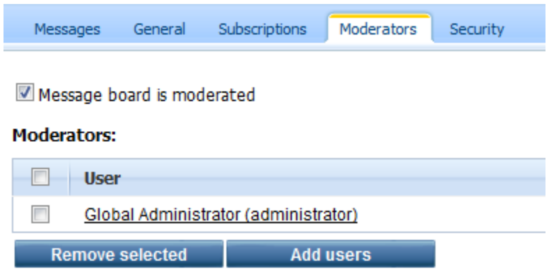
Subscriptions tab

On this tab, you can see a list of subscriptions to receiving notifications about new board messages. You can create new subscriptions using the **New subscription** link. Displayed subscriptions can be filtered by **E-mail** address and **User name**. You can also **Edit** ( ) or **Delete** ( ) subscriptions in the list.



Moderators tab

On this tab, you can make the message board moderated by checking the **Message board is moderated** check-box. In such case, new messages will be displayed only after approval by some of the moderators listed in the **Moderators** list-box below. Moderators can be **Added** or **Removed** using the corresponding buttons.



Security tab

On this tab, you can set security-related properties of the message board.

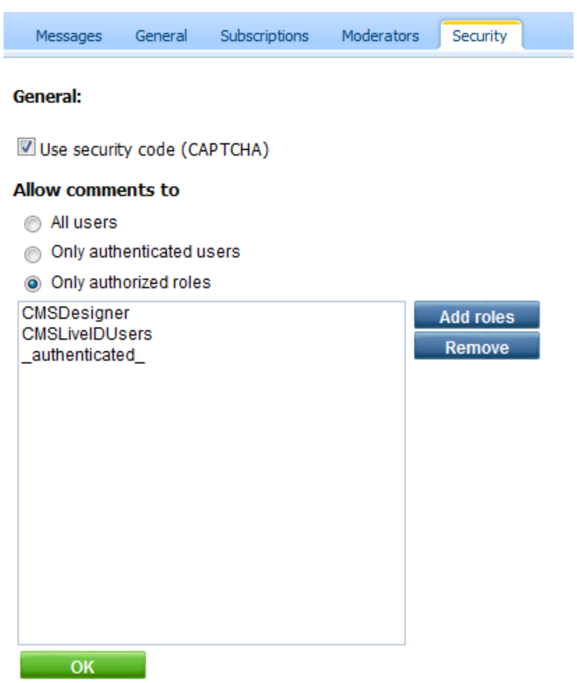
If the **Use security code (CAPTCHA)** check-box is checked, users will have to retype the CAPTCHA security code before adding a new message.

The **Allow comments to** section can be used to define which users can add new messages to the board. The following options are available:

- **All users** - everyone can add messages to the board
- **Only authenticated users** - only signed-in users can add messages to the board
- **Only authorized roles** - only members of the roles in the list-box below can add messages to the board

The following two options can be set only for group message boards:

- **Only group members** - only members of the group can add messages to the board
- **Only group admin** - only administrators of the group can add messages to the board



8.26.5 Setting Board base URL

Board base URL is the URL which will be used as the URL base for unsubscription links in notification e-mails. It can be set in two ways:

- In **Site Manager -> Settings -> Message boards -> Board base URL**; from here, it can be inherited by the web parts.
- Directly in **Message board web part properties.**

The following rules should be followed when creating user and group message boards placed on these pages in order for the unsubscription links to work correctly:

1. When you create a **user message board**, which is a message board placed on a user's profile, it is recommended to set the **Board base URL** directly in web part properties, for example like this:

```
~/Members/Profile.aspx
```

You can find a live example of this setting on the Community Site sample website.

2. When you create a **group message board**, which is a board placed on a group's profile, it is recommended to set the **Board base URL** directly in web part properties, for example like this:

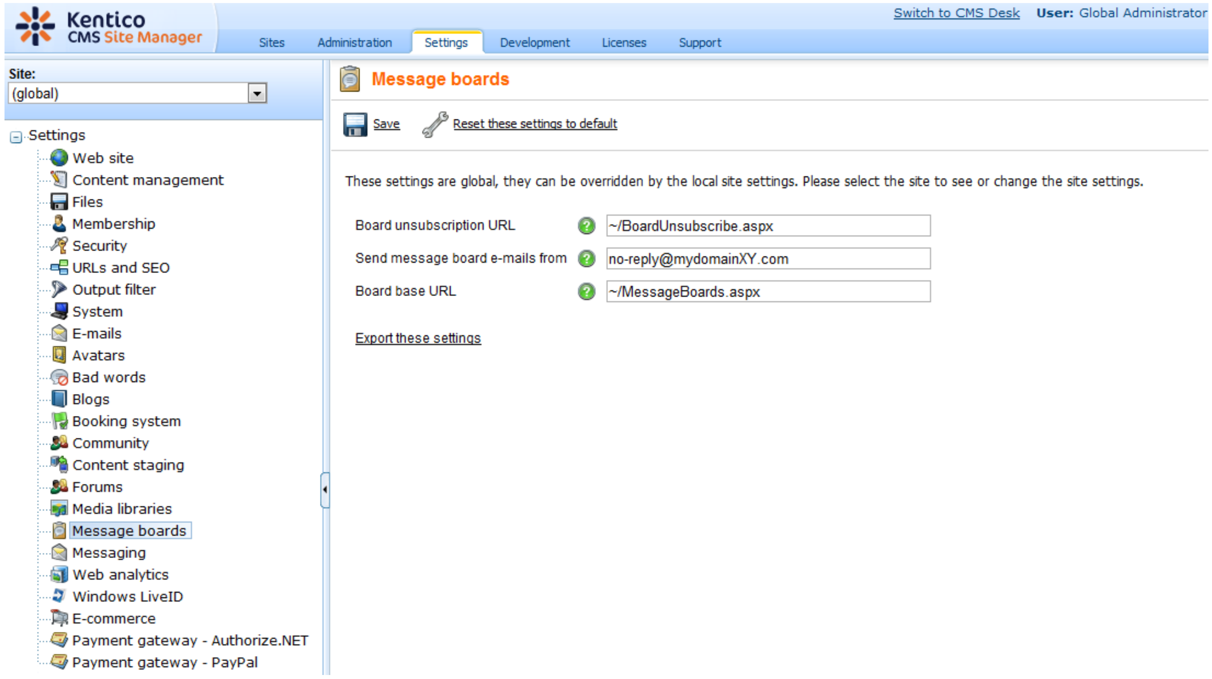
```
~/Groups/Profile.aspx
```

3. When you create a public message board, which is a board placed on any document without a wildcard URL, the Board base URL property needn't be set.

8.26.6 Settings

Settings of the **Message boards** module are located in **Site Manager -> Settings -> Message boards**. Among other community-related settings, the following settings are related to the **Message boards** module:

- **Board unsubscription URL** - URL of the site on which the **Message board unsubscription** web part is placed; this web part handles requests for unsubscription from notifications about new message board messages
- **Send message board e-mails from** - e-mail address that will appear in the From field of notification messages about new message board messages
- **Board base URL** - global board base URL that can be inherited by message boards; it can be used by board notification e-mails and message board viewers



8.26.7 Security

Message board

Based on the **Access** and **Message board owner** properties of the **Message board** web part, you can determine who will be allowed to add new messages to the board.

> **Changing the values**
>
> Remember that once the message board is created (after first message or subscription), you cannot make changes to these settings in the **New board settings** section of web part properties. You can only modify values of the **Access** property on the **Security** tab when editing the corresponding message board in **CMS Desk -> Tools -> Message boards**.

The following table explains **who can add messages to the board** under particular configurations. The difference between **User** and **Public boards** is that **Public boards** are always related to a document, while **User boards** are always related to a document and a user.

Public boards are typically used when you want multiple users to post messages to it. **User boards** are typically used on user profiles, as you can see on the **Community starter site** sample website, on the **Members -> Profile** page.

Message board owner	Access	Anonymous user	Authenticated user	Authorized role	Owner	Owner in authorized role

Public board	**All users**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Public board	**Authenticated users**		**Yes**	**Yes**	**Yes**	**Yes**
Public board	**Authorized roles**			**Yes**		**Yes**
Public board	**Owner**				**Yes**	**Yes**
User	**All users**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
User	**Authenticated users**		**Yes**	**Yes**	**Yes**	**Yes**
User	**Authorized roles**			**Yes**		**Yes**
User	**Owner**				**Yes**	**Yes**

When a board is in the **User x Owner** configuration, the following conditions need to be met in order for the current user to be able to post messages:

- the page must be accessed with *userid* or *username* parameter in querystring
- the current user must be the same as the one whose *userid* or *username* is passed in querystring
- the current user must not be hidden (configured by the *Is hidden* option when editing the user)

This can be typically used on user profiles, where messages to such board can be posted only by the owner of the profile, while other users can only read these messages. An example of such board is the **MessageBoardAnnouncements** board on the **/Members/Profile** page of the sample Community Site.

Group message board

The Group message board is always related to some group, hence only the **Access** property can be set. You can see a typical usage of this web part on the **Community starter site** sample website, on the **Groups -> Profile** page.

Access	**Anonymous user**	**Authenticated user**	**Authorized role**	**Group member**	**Group member in authorized role**	**Group admin**
All users	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Authenticated users		**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Authorized roles			**Yes**		**Yes**	**Yes**
Group members				**Yes**	**Yes**	**Yes**
Group admin						**Yes**

When the **Group members** option is set, the following conditions need to be met in order for the current user to be able to post messages:

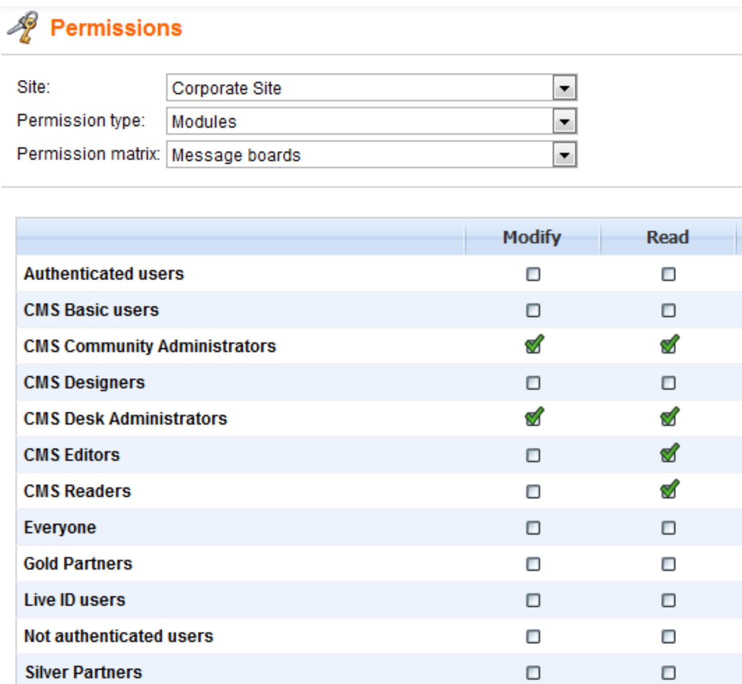
- the page must be accessed with *groupid* parameter in querystring
- the current user must be member of the group whose *groupid* is in the querystring
- the current user must not be hidden (configured by the *Is hidden* option when editing the user)

This can be typically used on group profiles, where messages to such board can be posted only by members of the group, while other users can only read these messages. An example of such board is the **GroupMessageBoard** board on the **/Group-pages/<group name>** page of the sample Community Site.

Permissions

Permissions for access to **Message boards** administration interface can be set in **Site Manager -> Administration -> Permissions**. You have to select the **Modules -> Message boards** permission matrix.

- **Modify -** members of the roles are allowed to edit message board settings, delete the boards and manage message board posts
- **Read -** selected role members are allowed to read the records and configuration of particular message boards, but are not allowed to modify them



8.26.8 Message board notifications

8.26.8.1 Who can be notified

When a new board message is added or during message editing, notifications can be sent to:

- **Board moderators**
- **Subscribers**

The following text explains to whom notifications are sent under specific conditions:

New message

1. has been added by the board moderator, global administrator, board owner (in case of a user board), user with the **Modify** permission or group administrator (in case of group boards)

- the message is marked as APPROVED
- the e-mail is sent to subscribers

2. has been added by anybody else
a) board is moderated

- the message is marked as NOT APPROVED
- the e-mail is sent to moderators

b) board is not moderated

- the message is marked as APPROVED
- the e-mail is sent to subscribers

Message editing

1. the message is switched from NOT APPROVED to APPROVED

- the e-mail is sent to subscribers

2. other message changes

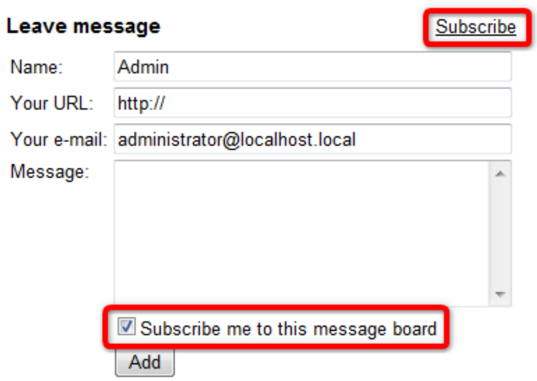
- no e-mail is sent

8.26.8.2 User subscriptions

Message board notifications

You can let users subscribe to receiving notifications about new messages added to the message board. You need to check the **Enable subscriptions** check-box in **Message board** web part properties. Alternatively, when the message board is already created, you can allow subscriptions in **CMS Desk -> Tools -> Message boards -> Boards**. If you choose to **Edit** () the message board and switch to the **General** tab, you can check the same **Enable subscriptions** check-box here, which enables the subscriptions.

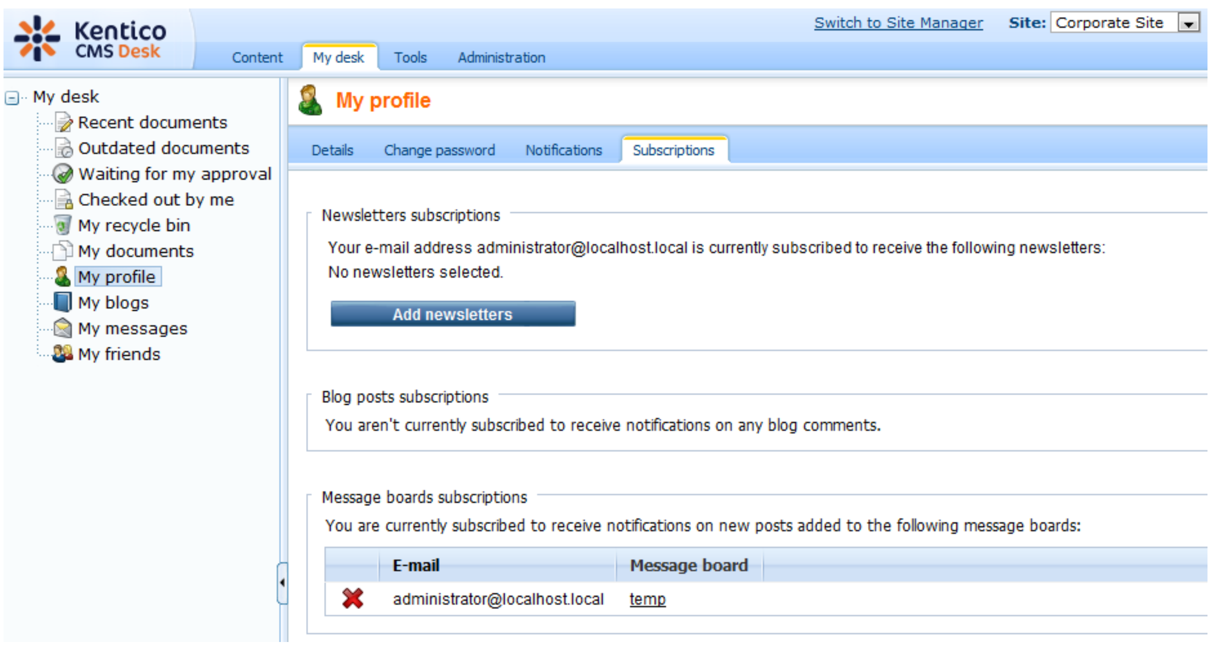
The subscription itself can be done two ways. Users can either check the **Subscribe to message board** check-box, which subscribes them along with adding the message. They can also click the **Subscribe to board** link, which displays only one **E-mail** field. After entering their address and clicking the **Subscribe** button, they can subscribe to notifications about this board without leaving any message.



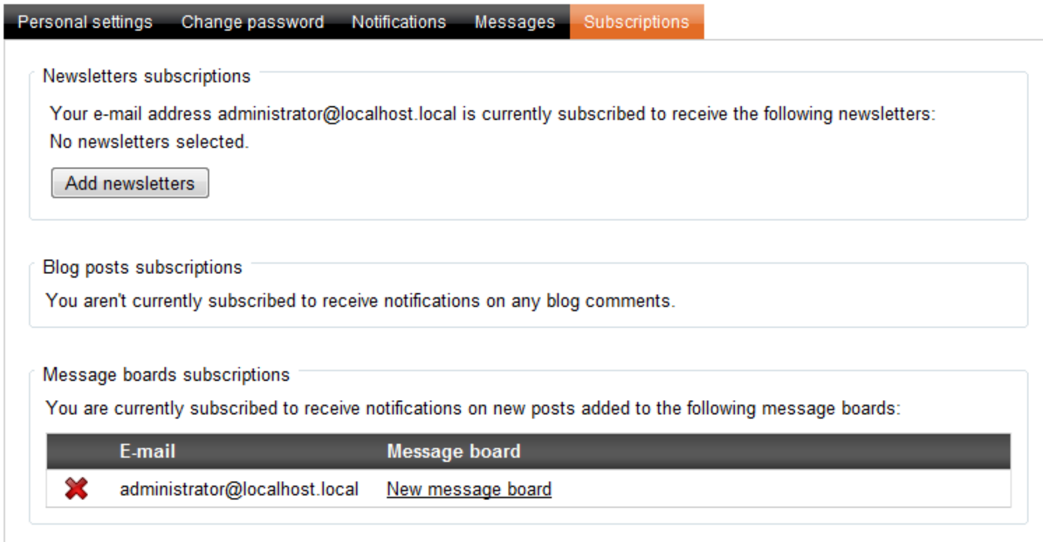
Subscriptions management

Users can view their subscriptions and eventually **unsubscribe** using the **Delete** () icon at the following two places:

1. Users with access to **CMS Desk** can view their subscriptions on the **My Desk -> My profile -> Subscriptions** tab.

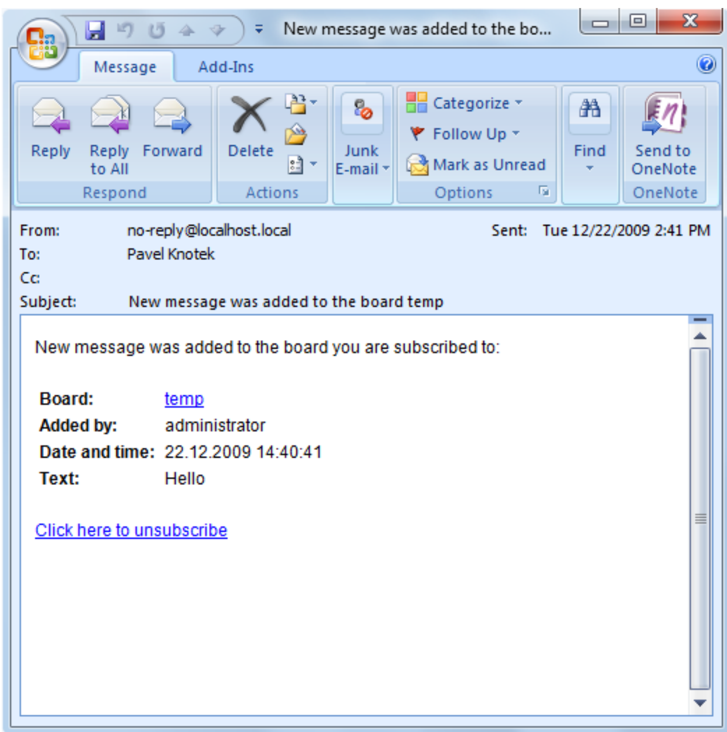


2. On live site, users can view their subscriptions in the **My account** web part. The **Display my subscriptions** property of the web part must be enabled for this to be possible.



Unsubscription links configuration

Each message board notification e-mail contains an unsubscription link. By clicking the link, users can unsubscribe from receiving notifications about new messages.



For the unsubscription links to work, you have to do the following tasks:

1. Place the **Message board unsubscription** web part to a page. It is recommended to create a special page for this purpose, as you can see at **Corporate site -> Special pages -> Board unsubscribe** or **Community site -> Special-pages -> Board unsubscribe**. You can set only one property of the web part - **Confirmation text**. This is the text that will be displayed after successful unsubscription.

2. Set the URL of the page created in step 1 as the **Unsubscription URL** property of the message board. This can be done three ways:

- Enter the URL into the **Board unsubscription URL** field in **Site Manager -> Settings -> Message boards**. This URL will be used by default when no URL is entered in web part properties of the message board.
- When adding the **Message board** web part, you can set set its **Unsubscription URL** property. This setting overrides the option in **Site Manager -> Settings -> Message boards**.
- When the message board is created, you can edit the **Unsubscription URL** property on the **CMS Desk -> Tools -> Message boards -> Boards** tab **-> Edit (****)** the message board **-> General** tab.

8.26.8.3 E-mail templates

There are two different e-mail templates that can be used when sending notifications to subscribers and board moderators:

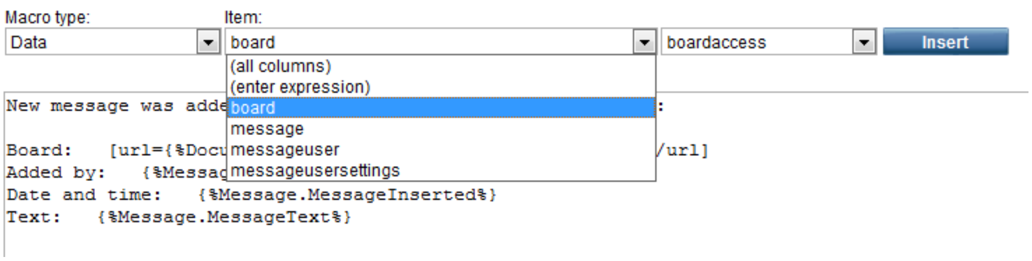
- **Subscribers** - e-mails are based on the **Boards - Notification to board subscribers** template
- **Moderators** - e-mails are based on the **Boards - Notification to board moderators** template

The following macros can be used in the notification e-mails:

Data macros

- **Board.XXX** - where XXX are columns of the *Board_Board* table
- **Message.XXX** - where XXX are columns of the *Board_Message* table
- **MessageUser.XXX** - where XXX are columns of the *CMS_User* table
- **MessageUserSettings.XXX** - where XXX are columns of the *CMS_UserSettings* table

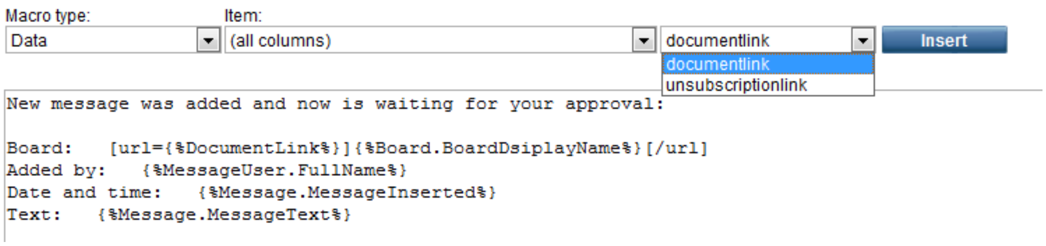
Example: *{%Message.MessageEmail%}*



Source macros

- **DocumentLink** - link to the document where the board is placed
- **UnsubscriptionLink** - unsubscription link

Example: *{%DocumentLink %}*



8.27 Messaging

8.27.1 Overview

The **Messaging** module allows users of your website to communicate via text messages. Its purpose is to provide an internal way of communication with other users of the website, both on the live site and in the user interface.

Users with access to Kentico CMS user interface can send, receive and manage their messages in **CMS Desk -> My desk -> My messages**. Functionality provided by this section of the user interface is described in the My messages topic. On the live site, identical functionality can be provided by a single **My messages** web part, or separately by other web parts which come with the module. All Messaging module web parts are described in the Adding the messaging functionality to the live site topic.

It is possible to let Kentico CMS notify users about new messages by means of automatic notification e-mails. Information related to this topic can be found in the E-mail notifications topic. The Security topic explains messaging possibilities of unregistered anonymous website visitors.

The Messaging internals and API sub-chapter provides an overview of database tables and API classes used by the module. It also provides examples of how methods from the classes can be used to handle messaging in your custom code.

There are other ways how you can enable users of your website to communicate with each other. The Forums module enables you to add standard discussion forums to the live site. The Message boards modules provides similar functionality as the forums, with the difference that message boards are not structured and are typically used to let users add comments on the content of a particular page.

8.27.2 My messages

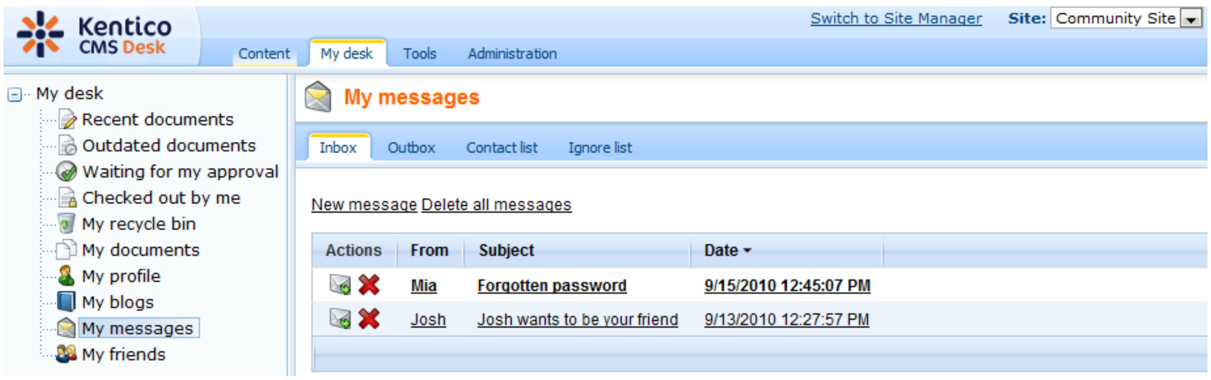
The **CMS Desk -> My desk -> My messages** section of the administration interface is divided into the following four tabs:

Inbox

On this tab, the current user can see all their received messages. A particular message can be displayed by clicking its **Subject**, **Date** or sender in the **From** column. The following two actions are available for each message:

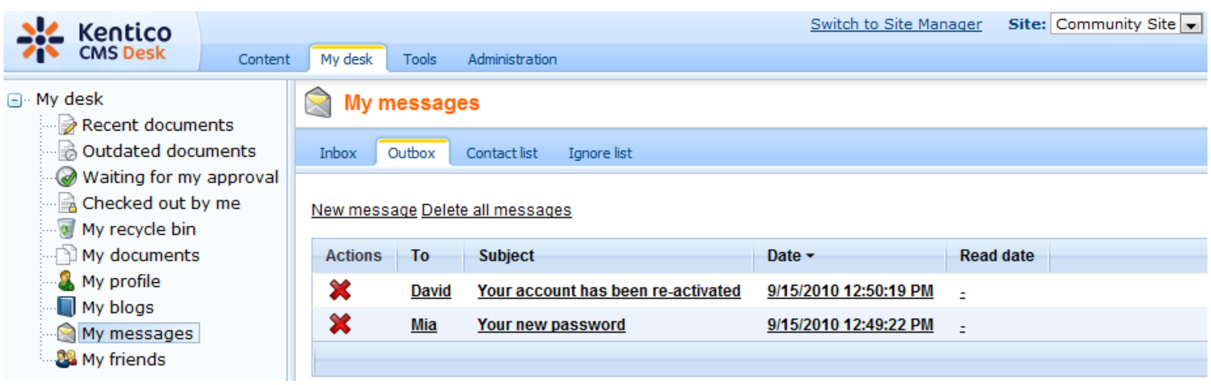
- **Reply** - opens a dialog where a reply to the message can be written and sent out
- **Delete** - deletes the particular message

The **Delete all messages** link deletes all messages in the list.



Outbox

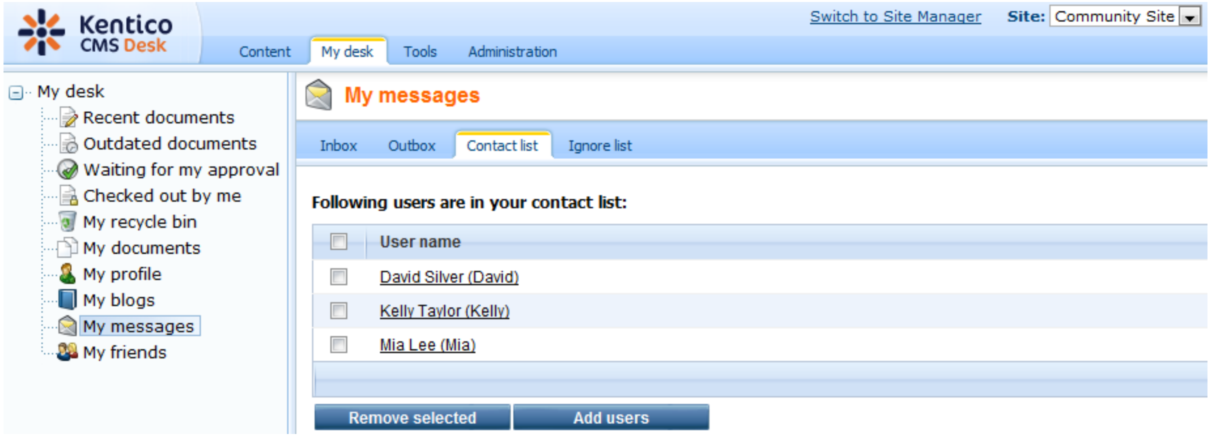
This tab lists all messages sent out by the current user. A particular message can be displayed by clicking its **Subject**, **Date** or recipient in the **To** column. You can delete particular messages using the **Delete** ( ) icon, or delete all listed messages using the **Delete all messages** link. The **New message** link opens a dialog where a new message can be written and sent out.



Contact list

This tab represents the current user's contact list, i.e. a list of users they communicate with. Having a user in the contact list makes it easier to select the user as a recipient when creating a new message.

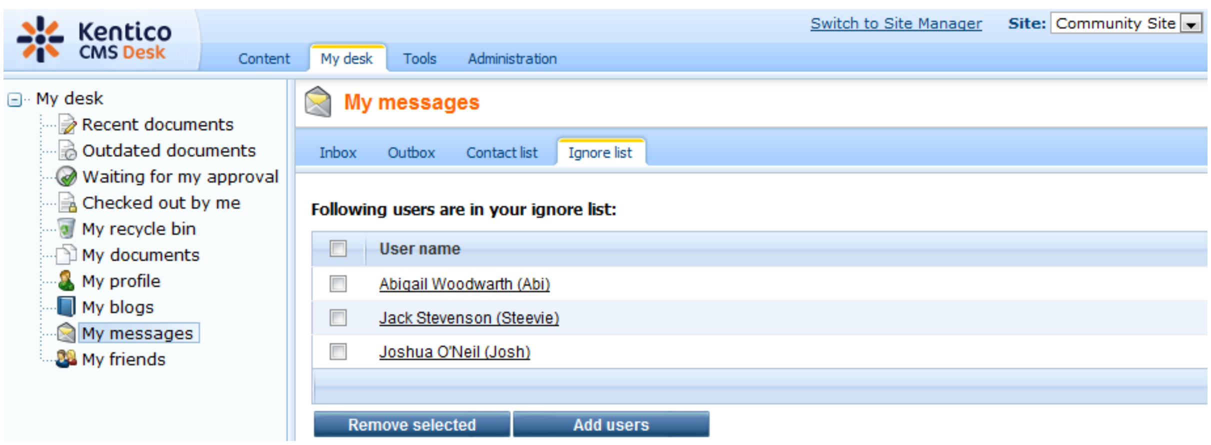
The **Add users** button opens a pop-up dialog where you can select users to be added to the list. The **Remove selected** button removes users selected by the check-boxes from the list.



Ignore list

This tab represents the current user's ignore list, i.e. a list of users from which no messages will be received. If a user who is in your ignore list sends you a message, the message will simply not be delivered.

The **Add users** button opens a pop-up dialog where you can select users to be added to the list. The **Remove selected** button removes users selected by the check-boxes from the list.



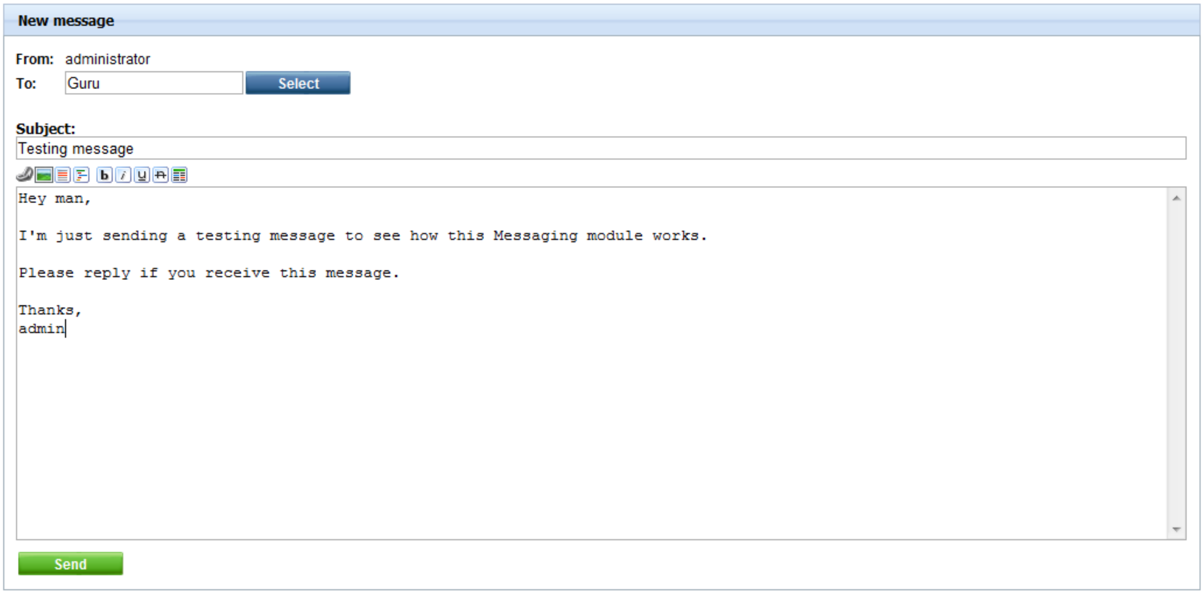
New message

On the **Inbox** and **Outbox** tabs, you can find the **New message** link. Clicking the link opens the **New message** dialog where new messages can be created and sent out. The **New message** dialog is depicted in the screenshot below.

When creating a message, you first need to select its recipient by clicking the **Select** button next to the **To** field. Clicking the button opens a pop-up dialog where the recipient can be selected from the contact list, from the user's friends, or searched from all visible website users.

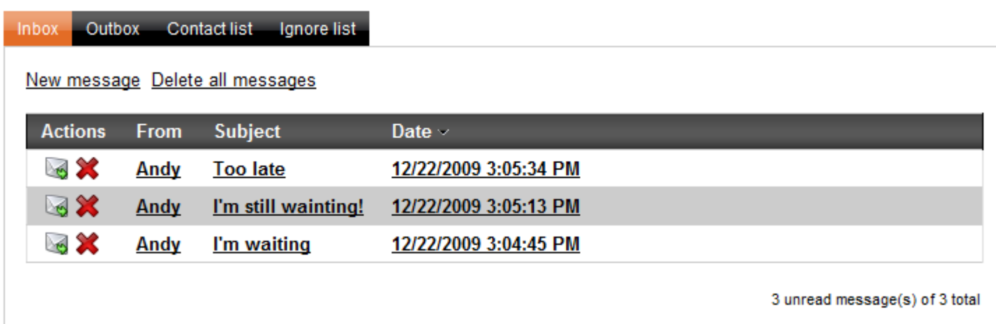
Then you need to fill in the **Subject** field and the actual text of the message into the large text area. Entered text can be formatted using BBCode. This is possible either by entering BBCode tags directly, or using the icons above the main text area. Supported BBCode tags are listed and explained in Modules -> Forums -> BBCode support.

Finally, the message can be sent out using the **Send** button.



8.27.3 Adding the messaging functionality to the live site

The same functionality that is described in the My messages topic can be added to the live site as well. It is possible using the web parts stored under the **Messaging** web part category. This page gives just a brief overview of the messaging web parts. Detailed descriptions of each web part and its properties can be found in Kentico CMS Web Parts Reference.



The **My messages** web part, shown in the screenshot above, is an all-in-one messaging web part. It provides exactly the same functionality as the user interface in **CMS Desk -> My Desk -> My messages** . This web part encapsulates the rest of the messaging web parts in one web part. If you need to add just a particular part of the messaging functionality,you can use one of the other messaging web parts:

- **Contact list** - allows users to add other users into their contact list or remove them from the list
- **Ignore list** - allows users to define which users they don't want to receive any messages from
- **Inbox** - shows a list of received messages and allows replying to them and deleting them
- **Messaging info panel** - displays links to inbox, outbox, etc.
- **Outbox - s**hows a list of sent messages and allows deleting them
- **Send message** - allows users to send messages

These web parts can be used separately on any page of the website, providing the same functionality as when the **My messages** web part is used. You can also choose which of these web parts will be included in the **My messages** web part. This can be done using its **Display inbox**, **Display outbox**, **Display contact list** and **Display ignore list** properties.

8.27.4 E-mail notifications

Users can be notified about new messages received via the Messaging module by means of notification e-mails. The e-mails are based on the **Messaging - Notification email** e-mail template.

For this to work, the target e-mail address must be filled into a user's **Messaging notification e-mail** field. This can be done several ways:

- On the live site, it can be entered via the **My profile** web part.
- Users with access to the system's user interface can enter it in **CMS Desk -> My desk -> My profile**.
- Global administrators can edit this field of each user in **Site Manager -> Administration -> Users -> edit ( ) a user -> Settings**.

When the e-mail address is entered into the field, a notification e-mail is sent to the address whenever the user receives a new message. This does not apply to messages received from users in the recipient's **Ignore list**.

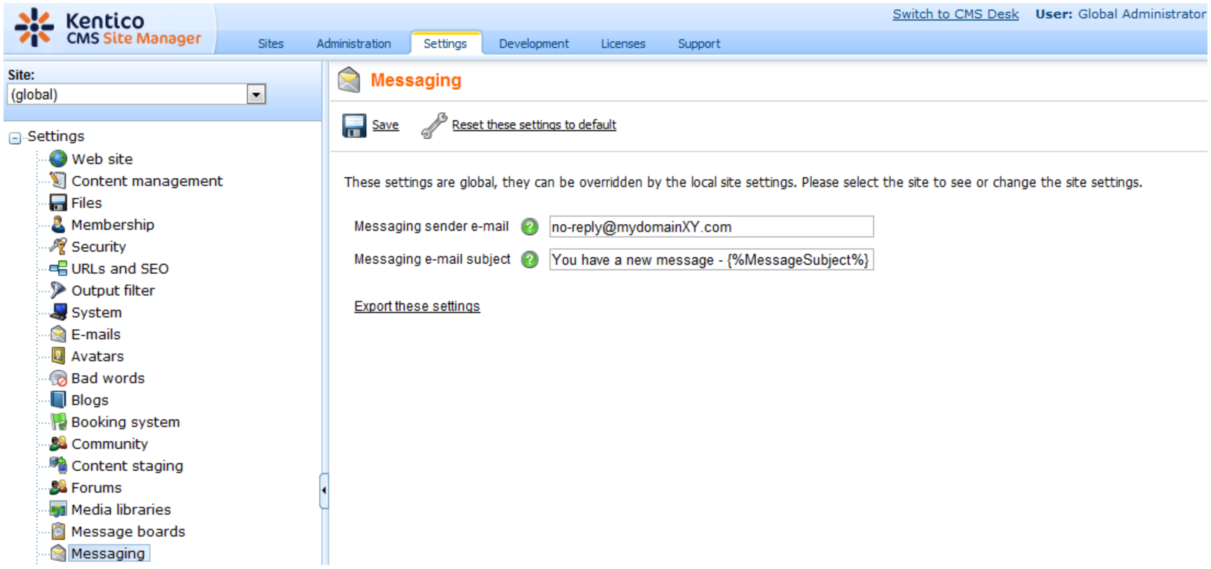
Please note

Your instance of Kentico CMS must be configured to use an SMTP server in order for e-mails to be sent, as described in Installation and deployment -> Additional configuration tasks -> SMTP server configuration.

Related settings

In **Site manager -> Settings -> Messaging**, you can find the following settings related to the notification e-mails:

- **Messaging sender e-mail** - e-mail address that will be used as the sender address (*From* field) of the notification e-mails
- **Messaging e-mail subject** - entered text will be used as content of the *Subject* field of notification e-mails



8.27.5 Security

Even **unregistered anonymous users** can send messages to registered users of your website. This is possible only from the **Send message** web part, the **My messages** web part can be used only by registered users.

To allow this functionality, go to properties of the **Send message** web part and check the **Allow anonymous users** check-box. By checking the **Allow anonymous users to select recipient** check-box below, you can give anonymous users permission to view a list of all registered users and select a user from this list.

8.27.6 Messaging internals and API

8.27.6.1 Database tables and API classes

The Messaging module uses the following database table:

- **Messaging_Message -** this is the database table used for storing messages
- **Messaging_ContactList -** this database table is used for storing contact lists; each record in the table contains an ID of a contact list owner in the first column and an ID of a user in the contact list in the second column
- **Messaging_IgnoreList -** this database table is used for storing ignore lists; each record in the table contains an ID of a ignore list owner in the first column and an ID of a user in the ignore list in the second column

The messaging API is provided by the **CMS.Messaging** namespace classes:

- **MessageInfo**, **MessageInfoProvider -** these classes provide functionality for managing messages

The following chapters show how to use methods from these classes in your custom code:

- [Creating a new message]
- [Deleting all messages]
- [Adding a user to contact list]

- Removing a user from ignore list

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.27.6.2 Creating a new message

The following sample code shows how you can create a new message, edit its text and finally delete it, all using the API.

[C#]

```
using CMS.Messaging;
using CMS.CMSHelper;
using CMS.SiteProvider;
using CMS.GlobalHelper;

...

        // Creates a new message from CurrentUser to administrator
        MessageInfo message = new MessageInfo();
        message.MessageSubject = "Subject";
        message.MessageBody = "Hello world!";
        message.MessageSenderUserID = CMSContext.CurrentUser.UserID;
        message.MessageRecipientUserID = UserInfoProvider.GetUserInfo(
"administrator").UserID;
        message.MessageSent = DateTime.Now;
        MessageInfoProvider.SetMessageInfo(message);

        // Gets the previously created message and edits its body
        MessageInfo messageToEdit = MessageInfoProvider.GetMessageInfo(message.
MessageID);
        messageToEdit.MessageBody += " This message was edited";
        MessageInfoProvider.SetMessageInfo(messageToEdit);


        // Deletes the previously created and edited message
        MessageInfoProvider.DeleteMessageInfo(message.MessageID);
```

8.27.6.3 Deleting all messages

The following sample code shows how you can delete all messages in current user's outbox using the API.

[C#]

```
using CMS.Messaging;
using CMS.CMSHelper;
using CMS.SiteProvider;
using CMS.GlobalHelper;

...

        // Deletes all messages in CurrentUser's outbox
        MessageInfoProvider.DeleteSentMessages(CMSContext.CurrentUser.UserID);
```

8.27.6.4 Adding a user to contact list

The following sample code shows how you can add a user to current user's contact list using the API.

[C#]

```
using CMS.Messaging;
using CMS.CMSHelper;
using CMS.SiteProvider;
using CMS.GlobalHelper;

...

   // Adds administrator to CurrentUser's Contact list
   UserInfo user = UserInfoProvider.GetUserInfo("administrator");
   ContactListInfoProvider.AddToContactList(CMSContext.CurrentUser.UserID, user.
UserID);
```

8.27.6.5 Removing a user from ignore list

The following sample code shows how you can remove a user from current user's ignore list using the API.

[C#]

```
using CMS.Messaging;
using CMS.CMSHelper;
using CMS.SiteProvider;
using CMS.GlobalHelper;

...

   // Removes administrator from CurrentUser's Ignore list
   UserInfo user = UserInfoProvider.GetUserInfo("administrator");
   IgnoreListInfoProvider.RemoveFromIgnoreList(CMSContext.CurrentUser.UserID,
user.UserID);
```

8.28 Newsletters

8.28.1 Overview

The Newsletters module allows the creation and management of highly configurable newsletters. Newsletters are used to regularly send e-mails about a certain topic to users who agree to receive them by subscribing.

When a newsletter issue is sent, e-mails are created and personalized (if applicable) for every individual subscriber. This procedure is performed by the Newsletter queue and any e-mails lost due to errors are automatically resent. This is handled by the **Send queued newsletters** scheduled task. Once this process is complete, the e-mails are sent either directly to the SMTP server or to the E-mail queue if the given newsletter is configured to do so (later topics in this chapter describe how this can be done). Using the e-mail queue is recommended for newsletters with large amounts of subscribers to ensure that all e-mails are delivered correctly.

The newsletters can be of two types:

- Static newsletters - every issue is edited and sent manually. The newsletters are based on predefined templates.
- Dynamic newsletters - issues are sent out to all subscribers automatically at a specified interval. The content is dynamically taken from a specified page, which is usually updated between newsletter issues.

Users can find information about newsletters and manage their subscriptions directly on the pages of a website if it contains the appropriate web parts. Please refer to the Integrating newsletters into the site topic to see what web parts are available.

Newsletters use templates to give all issues and related e-mails a unified appearance by using elements such as a company logo or footer. Templates may also contain various macros that are resolved into data matching individual recipients during the mail merge. For more details, please see the Newsletter templates topic.

The subscribers of all newsletters are stored in the system and can be monitored and managed as shown in the Subscriber management topic. The Subscriber import and export topic describes how tasks affecting large amounts of subscribers can be performed.

The Troubleshooting topic contains a list of solutions that can be used to resolve common problems with newsletter e-mails.

8.28.2 Creating a static newsletter

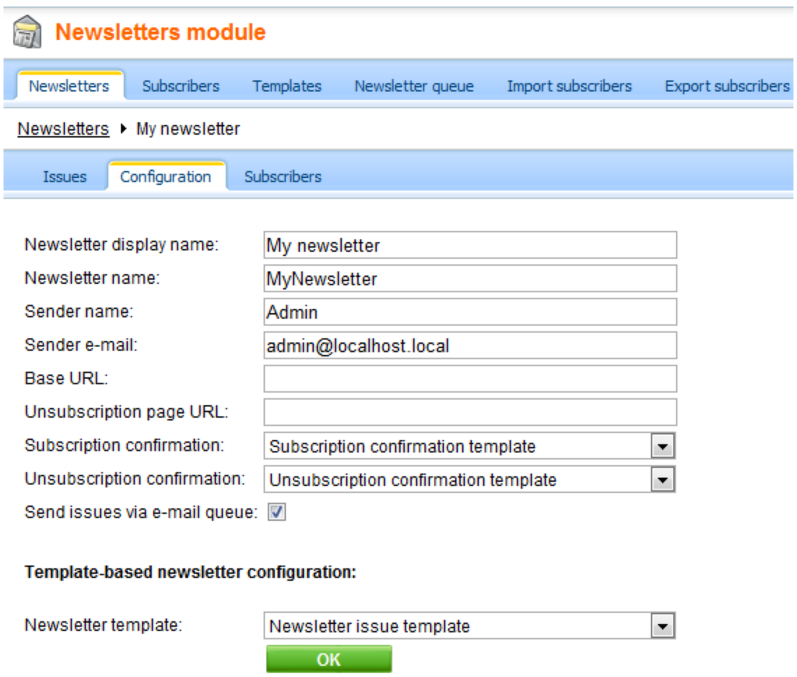
In this topic, you will learn how to create a static newsletter:

1. Go to **CMS Desk -> Tools -> Newsletter** and click **New newsletter**. Enter the following details:

- **Newsletter display name**: My newsletter *(this is the name displayed to the users)*
- **Newsletter name**: MyNewsletter *(this is the code name used in web parts and code)*
- **Subscription confirmation**: Subscription confirmation template
- **Unsubscription confirmation**: Unsubscription confirmation template
- **Sender name**: enter your full name
- **Sender e-mail**: enter your e-mail address

Choose **Template based newsletter** and use the **Newsletter issue template.** Click **OK**. The following three additional options can now be entered:

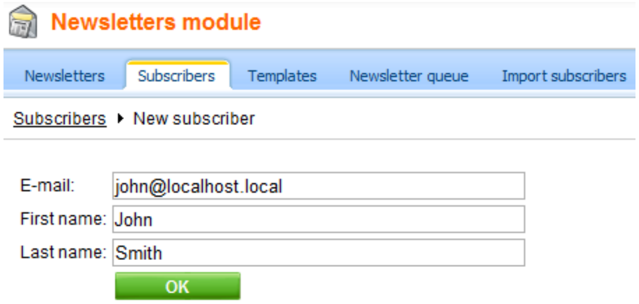
- **Base URL**: the base URL of your website used to convert relative links to absolute; It's necessary to set this property in order for the unsubscription links to work properly. It's also useful if you encounter any issues with links in newsletter e-mails - e.g. if you're using a different URL for your editing environment than for the live website. Example: https://www.example.com
- **Unsubscription page URL**: ~/SpecialPages/Unsubscribe.aspx (the page with a **Newsletter unsubscription** web part)
- **Send issues via e-mail queue**: if enabled, newsletter issues are sent to the SMTP server via the E-mail queue; this is recommended for newsletters with a large number of recipients; if disabled, the issues are sent directly to the SMTP server



Click **OK**.

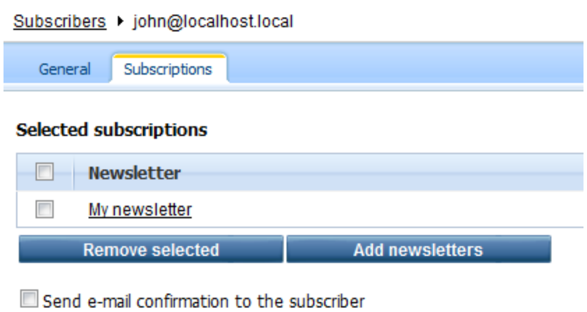
2. Now we will create a new subscriber. Go to the main **Subscribers** tab (in the top tab menu) and click **New subscriber**. Enter the following details:

- **E-mail:** subscriber's e-mail address
- **First name:** subscriber's first name
- **Last name:** subscriber's last name



Click **OK**. The subscriber has been created. Now we will assign this subscriber to our previously created newsletter.

3. Go to the **Subscriptions** tab, click the **Add newsletters** button, check the **My newsletter** checkbox and click **OK**.



If you check the **Send e-mail confirmation to the subscriber** checkbox, the subscriber will also receive an e-mail confirmation informing them of their subscription.

4. Your newsletter is configured. You can now create new issues as is described in the Authoring static newsletter issues topic.

8.28.3 Authoring static newsletter issues

Now that you have created a static newsletter with a subscriber as described in the Creating a static newsletter topic, it is possible to write and send individual issues. The following steps describe how this can be done:

1. Go to the **Newsletters** tab and edit ( ) **My newsletter**. Click **Create new issue** on the **Issues** tab. The wizard will guide you through the process of creating a new newsletter issue.

2. Enter the following **Subject**: Welcome to issue #1 of My newsletter

3. Now enter the following text into the **content** editable region:

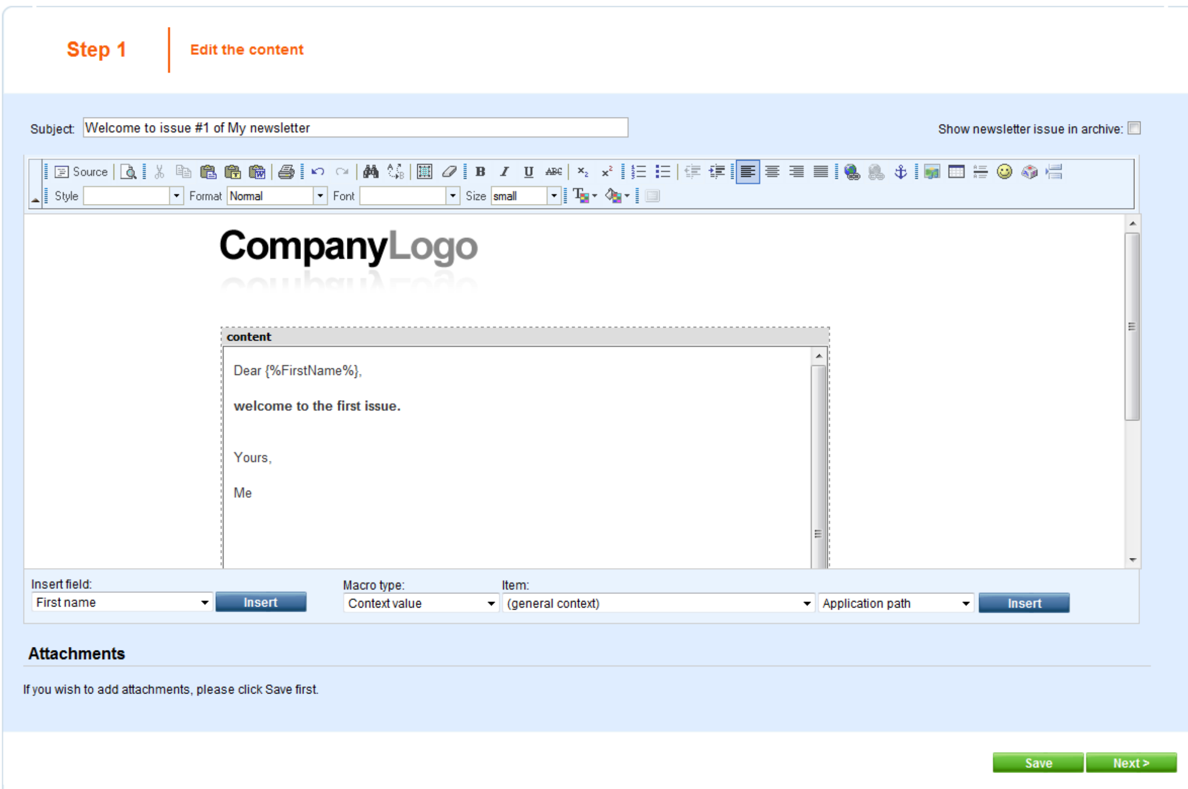
> Dear ,
>
> welcome to the first issue.

Yours,

Me

Place the cursor after the word Dear, choose **First name** from the **Insert field** drop-down list below the region and click the **Insert** button. The macro expression *{%FirstName%}* will be placed into the text. This macro is automatically replaced with the subscriber's first name during the mail merge.

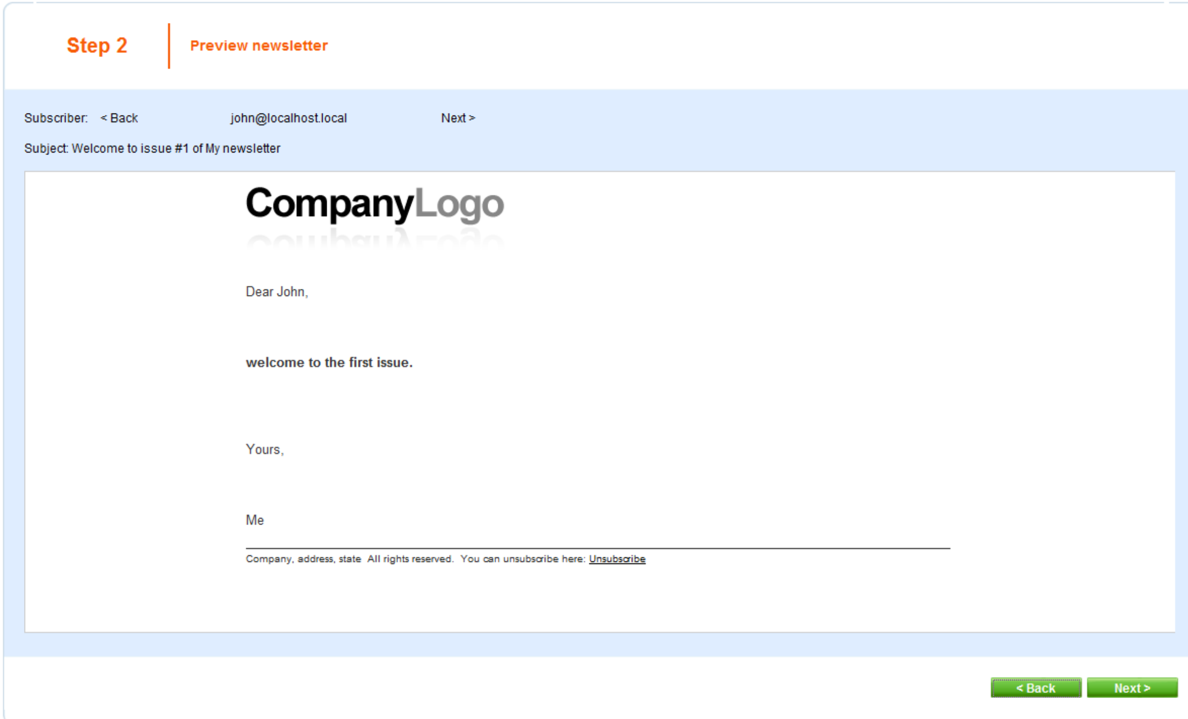
The issue now looks like this:



Newsletter templates

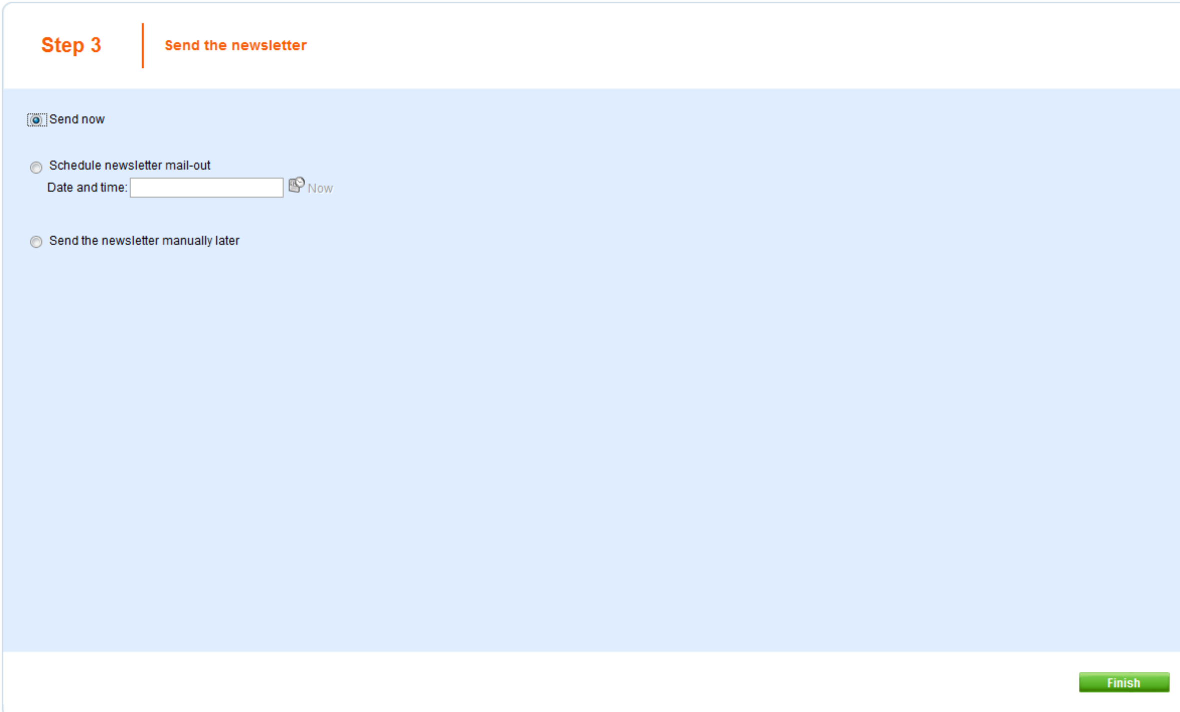
The structure of the newsletter text is defined by the newsletter issue template that can be edited on the **Templates** tab. Templates are described in the Newsletter templates topic.

4. Click **Next**. Now you can preview the content of the newsletter for each subscriber. You can click **Back** and modify the text if necessary.

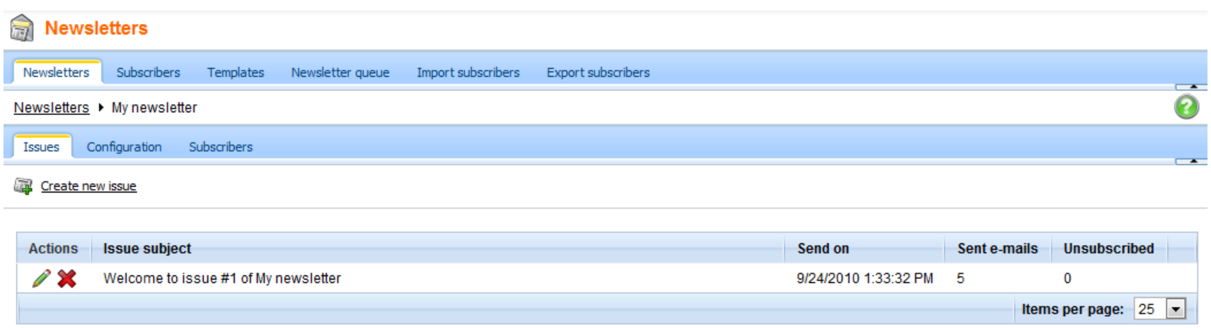


5. Click **Next** again. Now you can choose when the newsletter issue should be sent out:

- **Send now -** the newsletter issue is sent out immediately to all subscribers
- **Schedule newsletter mail-out -** the newsletter issue is sent out on a specified date and time
- **Send the newsletter manually later -** the newsletter is not sent and you can decide on the mail-out time later



Choose **Send now** and click **Finish**. On the Issues tab, you can see how many e-mails have already been sent out and how many subscribers unsubscribed after receiving this issue:



6. If you experience problems with receiving the newsletter issue e-mail, please follow the instructions in the Troubleshooting topic.

8.28.4 Creating a dynamic newsletter

Dynamic newsletters contain the content of a given page and they are sent out automatically on a regular basis, using the built-in scheduling system. The following steps will guide you through the creation of a dynamic newsletter:

1. Go to **CMS Desk -> Tools -> Newsletter** and click **New newsletter**. Enter the following details:

- **Newsletter display name**: My dynamic newsletter
- **Newsletter name**: MyDynamicNewsletter
- **Subscription confirmation**: Subscription confirmation template
- **Unsubscription confirmation**: Unsubscription confirmation template

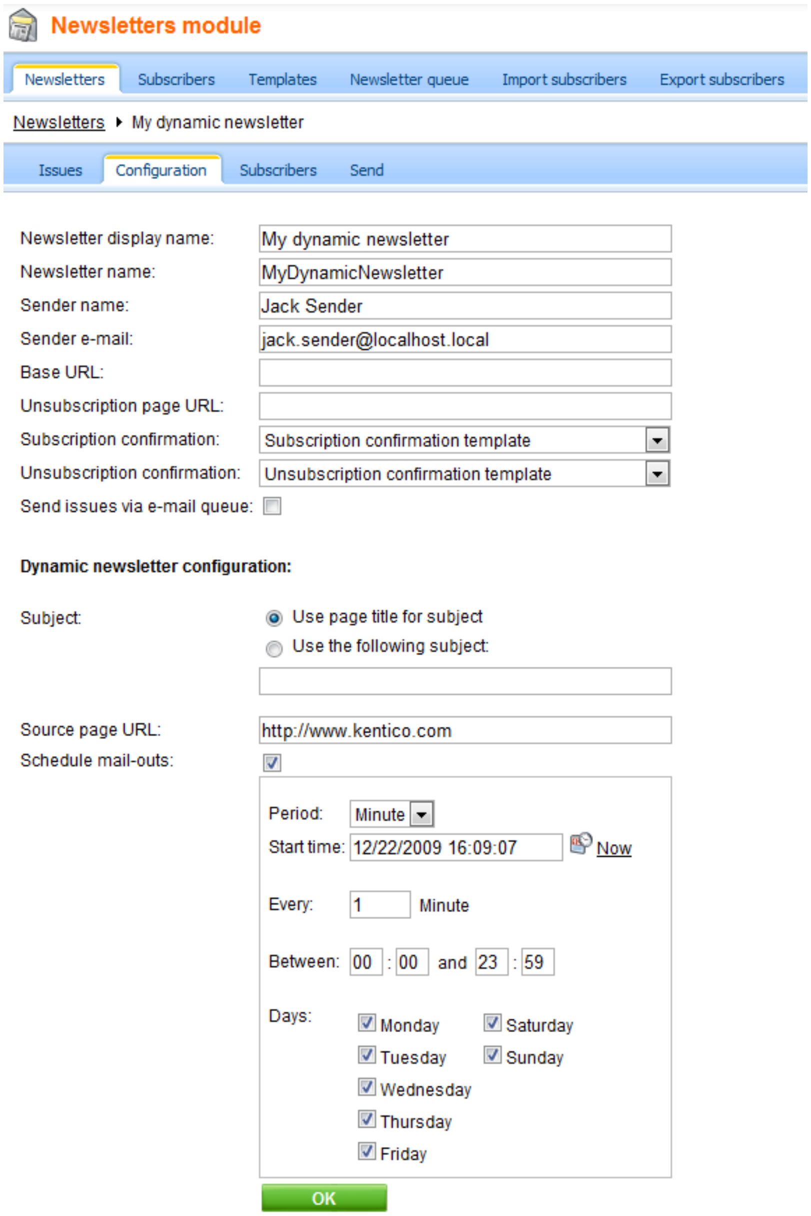
- **Sender name**: enter your full name
- **Sender e-mail**: enter your e-mail address

Choose **Dynamic newsletter** and enter the following details:

- **Source page URL**: *http://www.kentico.com/home.aspx* (this is the URL of the page the newsletter takes its content from)
- **Schedule mail-outs**: enabled
  - **Period**: Minute
  - **Start time**: use the date-time picker to select the current date and time (click **Now**)
  - **Every**: 1 minute
  - **Between**: 00:00 and 23:59
  - **Days**: check all days

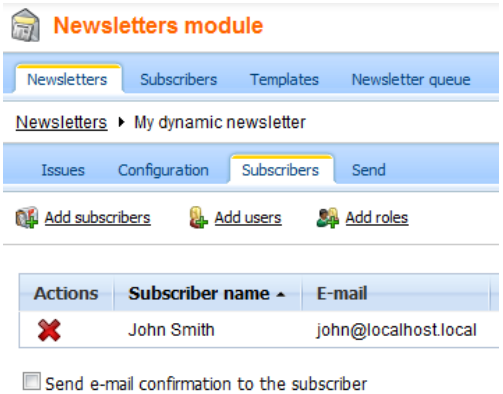
Click **OK**. The following three additional options can now be entered:

- **Base URL**: the base URL of your website used to convert relative links to absolute; It's necessary to set this property in order for the unsubscription links to work properly. It's also useful if you encounter any issues with links in newsletter e-mails - e.g. if you're using a different URL for your editing environment than for the live website. Example: https://www.example.com
- **Unsubscription page URL**: ~/SpecialPages/Unsubscribe.aspx (the page with a **Newsletter unsubscription** web part)
- **Send issues via e-mail queue**: if enabled, newsletter issues are sent to the SMTP server via the E-mail queue; this is recommended for newsletters with a large number of recipients; if disabled, the issues are sent directly to the SMTP server

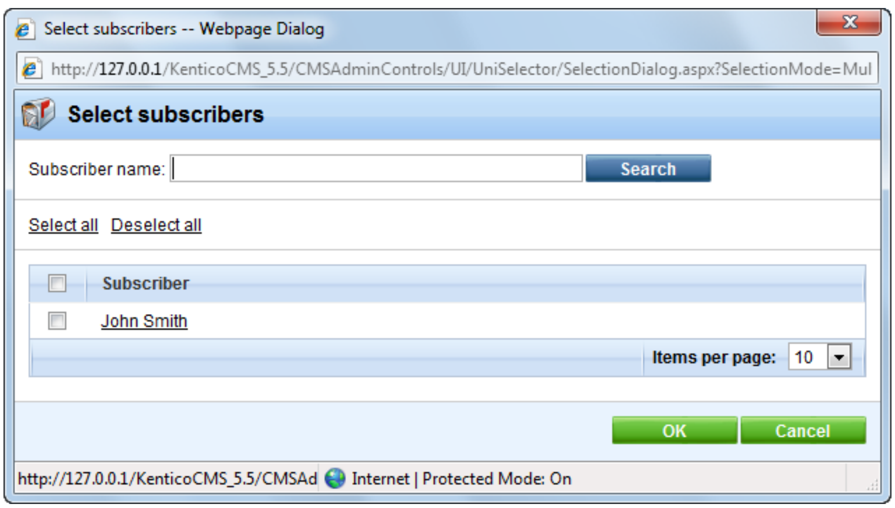


Click **OK**.

2. Now go to the **Subscribers** tab of this newsletter and add the subscriber created in the Creating a static newsletter topic to this newsletter.

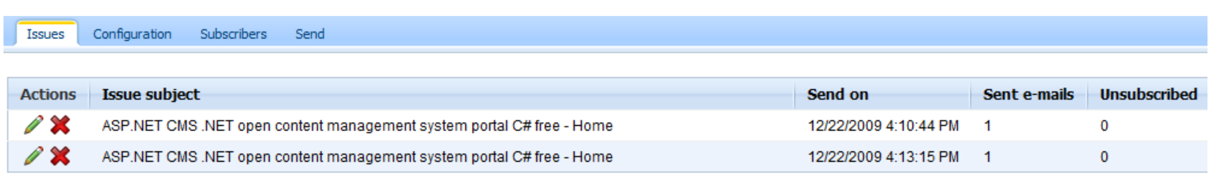


To do this, click the **Add subscriber** link, the following window will appear:



This window can be used for selecting from the subscribers of all newsletters. Alternatively, the **Add users** or **Add roles** links can be used to select from the users or roles of the current website. Make the selection and click **OK**. If you do this with the **Send e-mail confirmation to the subscriber** check-box checked, a notification e-mail will be sent to the selected users, informing them about the changes in their subscriptions.

3. Go to the **Issues** tab. Here, you will see the list of sent issues. You may need to wait up to 2 minutes until the first issue is sent out. You can refresh the page by clicking the **Issues** tab again.



4. Check your mail box, you should receive the content of the given page by e-mail:

In this way, you can send out any page from your website. You can create a new page only for newsletter purposes that will display e.g. new articles added to your website during the last month.

Blocking dynamic newsletter mail-out

If you want to block the mail-out of the page (e.g. if there are no new articles), you can either disable the **Schedule mail-outs** property of the dynamic newsletter on its **Configuration** tab or you can set the title of the source page to ##DONOTSEND## and the newsletter will not be sent. The title of a page can easily be changed at **CMS Desk -> Content -> Edit -> ... select a page from the content tree ... -> Properties -> Metadata -> Page title**.

8.28.5 Integrating newsletters into the site

You can integrate newsletters into your website using the following web parts:

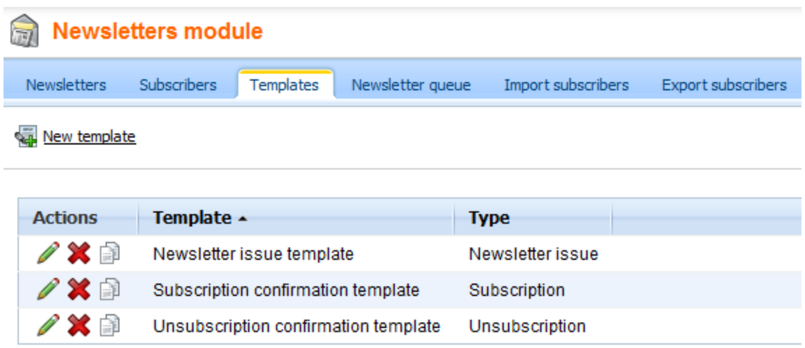
- **Newsletter subscription** - displays the newsletter subscription dialog
- **Newsletter unsubscription** - unsubscribes users and displays a confirmation message when the user clicks the "**Unsubscribe**" link in the newsletter issue
- **Newsletter archive** - displays archived issues of a specified newsletter
- **Unsubscription request** - this web part allows newsletter subscribers to request an unsubscription

e-mail (based on the **Newsletters - Unsubscription request** e-mail template) by submitting their e-mail address
- **My subscriptions** - this web part can be used by users to add or remove their newsletter subscriptions
- **My account** - if configured to display newsletter subscriptions, this web part can be used by users to add or remove their subscriptions

You can find more details on the properties of these web parts in **Kentico CMS Web Parts Reference**.

8.28.6 Newsletter templates

The e-mails sent by the Newsletter module are defined by templates. These templates can be managed at **CMS Desk -> Tools -> Newsletters -> Templates**.



There are three types of templates:

- **Newsletter issue** - this is a template that defines the layout and design of static newsletter issues. It contains editable regions where newsletter authors can enter the content.
- **Subscription** - template of the e-mail message sent when a user subscribes to the newsletter.
- **Unsubscription** - template of the e-mail message sent when a user unsubscribes.

All types of newsletter templates can contain the following macro expressions anywhere in the text:

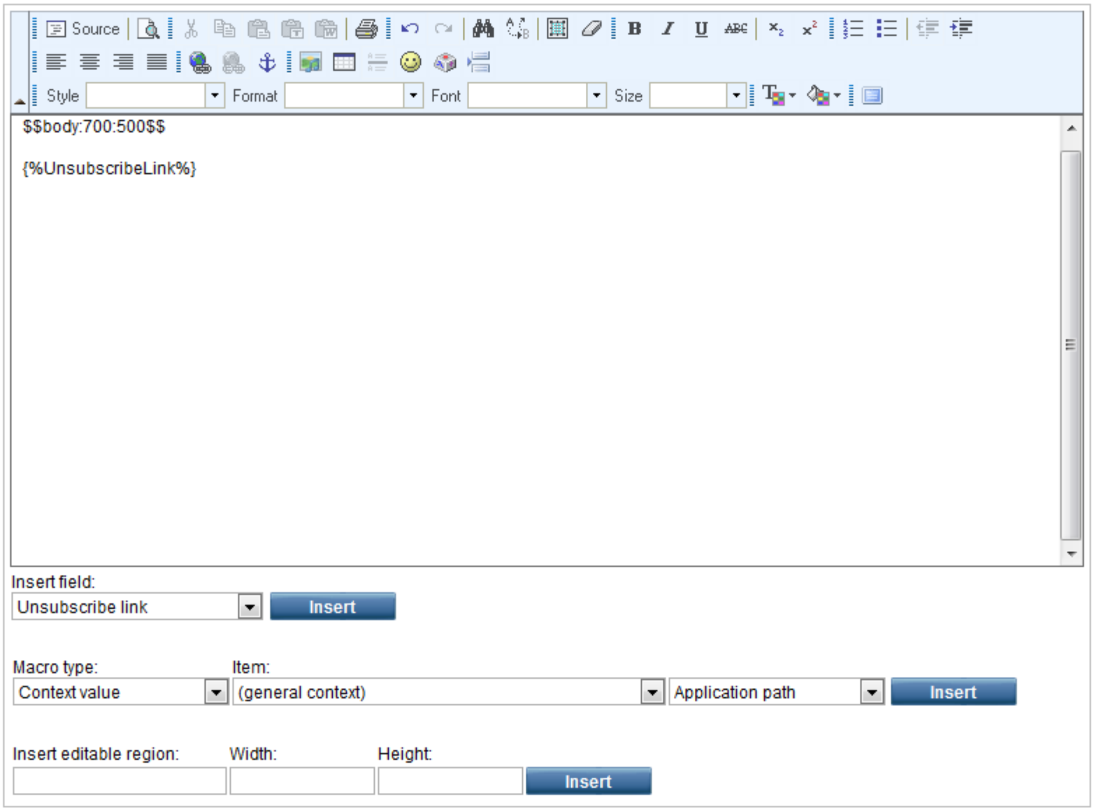
- **{%Email%}** - resolves into the e-mail address of the subscriber
- **{%FirstName%}** - resolves into the first name of the subscriber
- **{%LastName%}** - resolves into the last name of the subscriber
- **{%UnsubscribeLink%}** - resolves into a link to the unsubscription page

Additionally, other types of macro expressions as seen in Appendix A - Macro expressions can be entered into templates.

Editing newsletter issue templates

The issue templates consist of:

- **Header (HTML)** - the leading HTML code including the <html> element.
- **Body** - the layout of the e-mail - here you can enter static text, use the WYSIWYG editor and insert macros and editable regions. Editable regions allow users to enter the content of the particular newsletter issue. They can be inserted using the **Insert editable region** section:



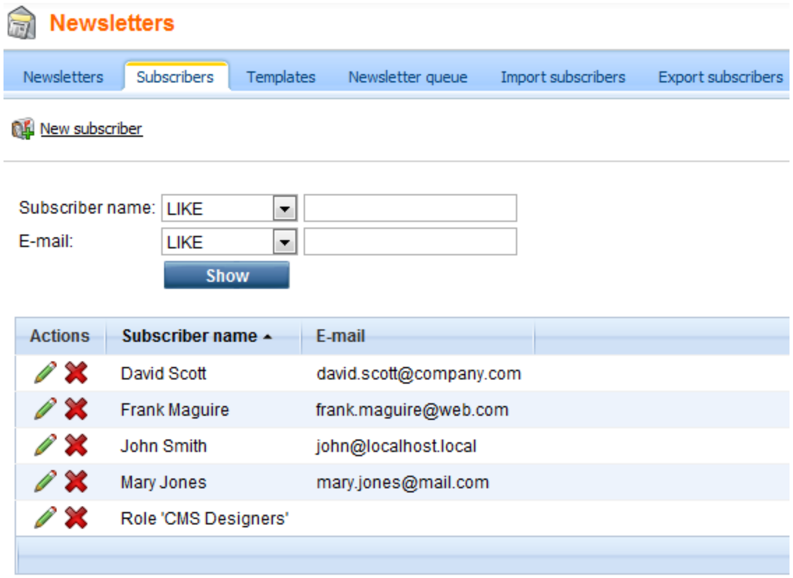
The region is inserted as a macro expression in format: *$$regionName:width:height$$*

- **Footer (HTML)** - the closing HTML code.
- **CSS stylesheet** - the CSS styles used for the newsletter; these styles are used by the newsletter issue editors and they are included in the e-mails.

File attachments can be added to the template at the bottom of the page.

8.28.7 Subscriber management

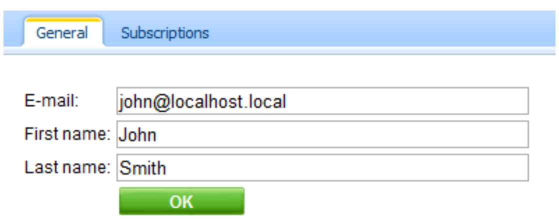
You can manage subscribers at **CMS Desk -> Tools -> Newsletters -> Subscribers**.



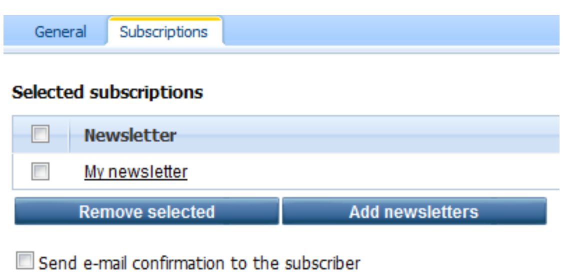
On this page, you can see a list of all subscribers of all of the website's newsletters. You can filter these by name or e-mail, by entering the desired value into the appropriate filter field and clicking the **Show** button.

By clicking the **Delete** () icon next to a subscriber record, you can remove the subscriber from the list and consequently from all newsletters they are subscribed to.

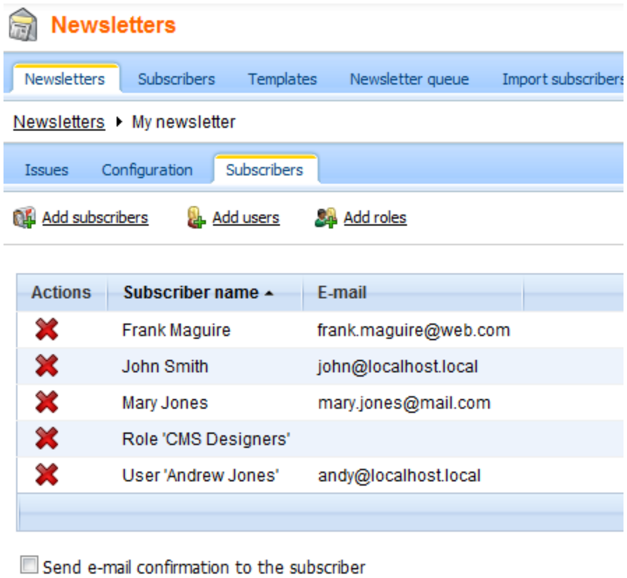
By clicking the **Edit** () icon next to a subscriber record, you can change the subscriber's details on the **General** tab. You have the same options here as when creating a new subscriber.



On the **Subscriptions** tab, you can select which newsletters this subscriber will be subscribed to. If you check the **Send e-mail confirmation to the subscriber** check-box, a notification e-mail will be sent to the user, informing them about the subscription changes.



Alternatively, subscribers of individual newsletters can be managed at **CMS Desk -> Tools -> Newsletters -> Newsletters -> ... edit () Newsletter ... -> Subscribers**. This tab displays a list similar to the one on the main Subscribers tab, but it only contains subscribers of the current newsletter. Clicking the **Delete** (✖) icon here doesn't completely remove a subscriber from the system, only from the current newsletter.



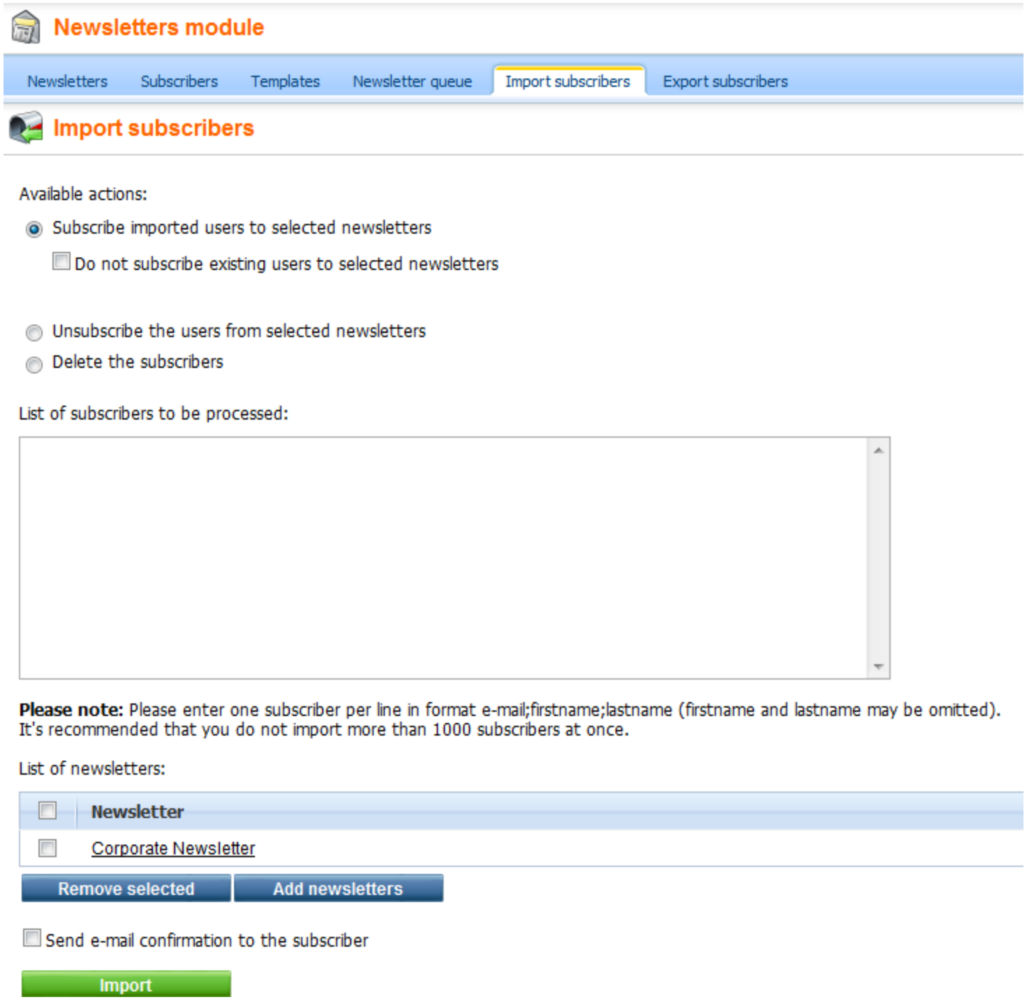
If you need to add or modify a large amount of subscribers, the **Import subscribers** tab described in the [Subscriber import and export](#) topic provides an easier way.

8.28.8 Subscriber import and export

When handling a large amount of subscribers, creating or modifying them one by one would be very slow and inefficient. The import and export functionality provides a way to perform mass actions or get subscriber data using specially formatted text.

Importing subscribers

The importing or modification of subscribers can be carried out using the **CMS Desk -> Tools -> Newsletters -> Import subscribers** dialog.



A list of subscribers needs to be prepared in the following format:

- **email;firstname;lastname**

Copy it to the '**List of subscribers to be processed**' text area. Each line should contain one record in the mentioned format. The following examples are all valid:

```
david.scott@company.com;David;Scott
mary.jones@mail.com;;Jones
frank.maguire@web.com;Frank
monica@italy.com
```

Clicking **Import** adds the subscribers from the list and performs the selected action. If one of the lines contains an invalid entry, the import is not processed for any of the records and an error is displayed. If a subscriber e-mail address already exists, the first name and last name are updated instead of creating a new subscriber.

By selecting one of the three radio buttons above the text area and adding some newsletters using the **Add newsletters** button below, the following actions can be done:

- **Subscribe imported users to the selected newsletters** - subscribes imported users to the selected newsletters
  - **Do not subscribe existing users to selected newsletters** - if checked, users with already existing e-mail addresses will not be subscribed to the selected newsletters, only their names will be updated
- **Unsubscribe the users from selected newsletters** - entered users will be unsubscribed from the newsletters selected below
- **Delete the subscribers** - entered users will be deleted from the list of subscribers

Exporting subscribers

If you need to export the list of subscribers to some other application, you can export them using the **CMS Desk -> Tools -> Newsletters -> Export subscribers** dialog.

You can choose if you want to export all subscribers (do not specify any newsletter) or only subscribers of chosen newsletter(s). The subscribers are exported in format:

- **email;firstname;lastname**

Then you can copy and paste the list of subscribers from the textbox to your application.

8.28.9 Troubleshooting

Problems with e-mails

If you are not correctly receiving newsletter e-mails, please check the following:

1. The newsletter issues are sent out using a scheduled task that is executed every 1 minute by default. You may need to wait for up to 2 minutes before you receive newsletter issue e-mail. The scheduled task status can be checked at **CMS Desk -> Administration -> Scheduled tasks**.

2. Go to **CMS Desk -> Tools -> Newsletter -> Newsletter queue**. If some e-mail failed, you will find the error message here. After you resolve the technical issue, you can resend all failed e-mails by clicking **Resend all failed**.

3. If your newsletters are configured to use the e-mail queue, go to **CMS Site Manager -> Administration -> E-mail queue**. Any e-mails lost due to errors can be monitored and resent here.

4. Make sure you're using correct e-mail addresses.

5. Make sure the newsletter issues aren't being blocked by some anti-spam software.

6. Go to **Site Manager -> Settings -> E-mails** and make sure your SMTP server is configured correctly You can find some additional details on SMTP server configuration in the SMTP server configuration topic. You can also test e-mail settings at **Site Manager -> Administration -> System -> E-mail**.

7. Newsletter e-mail debugging might be helpful when solving problems with newsletter e-mails. To enable it, add the following keys to the *web.config* file of your web project:

```
<add key="CMSLogEmails" value="true"/>
<add key="CMSDebugEmails" value="true"/>
```

The first key enables logging of all sent e-mails to *<web root>/AppData/logemails.log*. The second key ensures that all sent e-mails are logged, but not actually sent. This is helpful when you need to test the functionality but do not want the testing e-mails to actually be sent.

Problems with unsubscription links

If you encounter problems with unsubscription link resolving, you should check that the **Base URL** property of the newsletter is set. This property can be set at **CMS Desk -> Tools -> Newsletter**, choose to **Edit** (✏) the newsletter and switch to its **Configuration** tab.

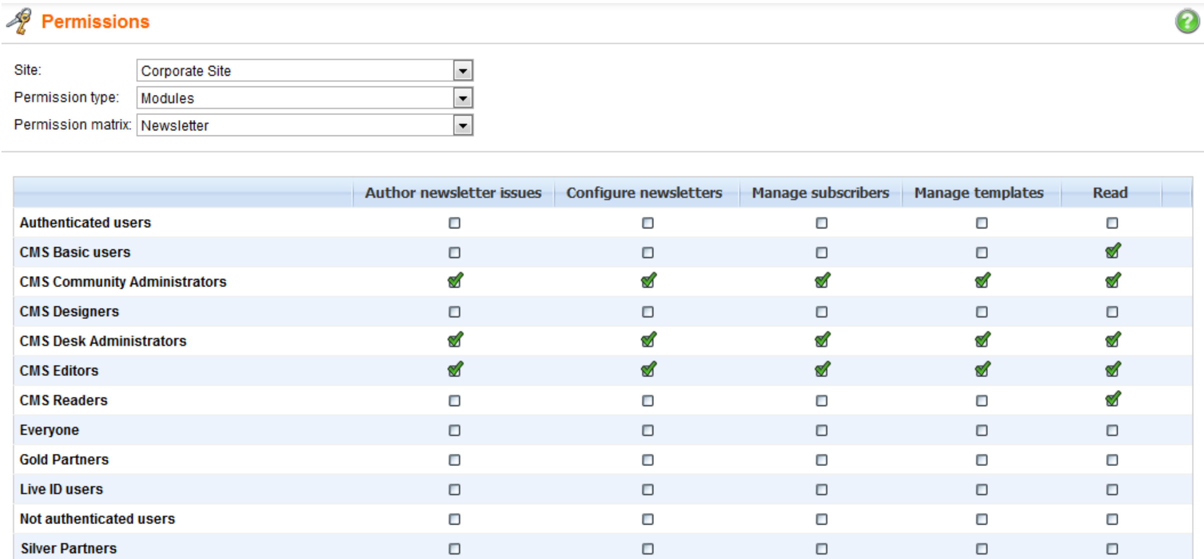
Problems with role subscribers

If you have a user role set as a subscriber for your newsletter, it is highly recommended to send the newsletter issues via the e-mail queue in order for the issues to be successfully sent to all members of the role. To set the newsletter to use the newsletter queue, go to **CMS Desk -> Tools -> Newsletter**, choose to **Edit** (✏) your newsletter, switch to its **Configuration** tab and enable the **Send issues via e-mail queue** property.

8.28.10 Security

You can control access to the Newsletters module from **CMS Site Manager (or CMS Desk) -> Administration -> Permissions**, after selecting the **Modules -> Newsletter** permission matrix. The Newsletter module has the following permissions:

- **Author newsletter issues** - members of the selected roles are allowed to create and edit newsletter issues
- **Configure newsletters** - members of the selected roles are allowed to configure newsletter settings
- **Manage subscribers** - members of the selected roles are allowed to add and remove newsletter subscribers
- **Manage templates** - members of the selected roles are allowed to create, edit and delete newsletter templates
- **Read** - members of the selected roles are allowed to view all data in the Newsletter module interface



8.29 Notifications

8.29.1 Overview

The Notifications module enables sending of notification messages using notification providers. Kentico CMS comes with a built-in e-mail notification gateway. However, you can create your own custom gateway such as an SMS or ICQ provider.

After creating a notification message template, site visitors or CMS Desk administrators can subscribe to receiving notifications about specific events, manage users' notifications and choose the providers from which they would like to receive notification messages.

Please note that for one event, users can obtain notification messages from multiple providers. There are three built-in events (document created, document updated, document deleted) and only the first of the three events is enabled by default. However, you can create and use your custom ones.

To create your own notification gateway, please refer to the Overview of the Custom notification gateway subchapter and learn how to:

- create a custom notification gateway form,
- create a custom notification gateway class,
- register a custom gateway,
- use the gateway on your site.

Finally, you can adjust the settings of the Notifications module and learn on its security possibilities.

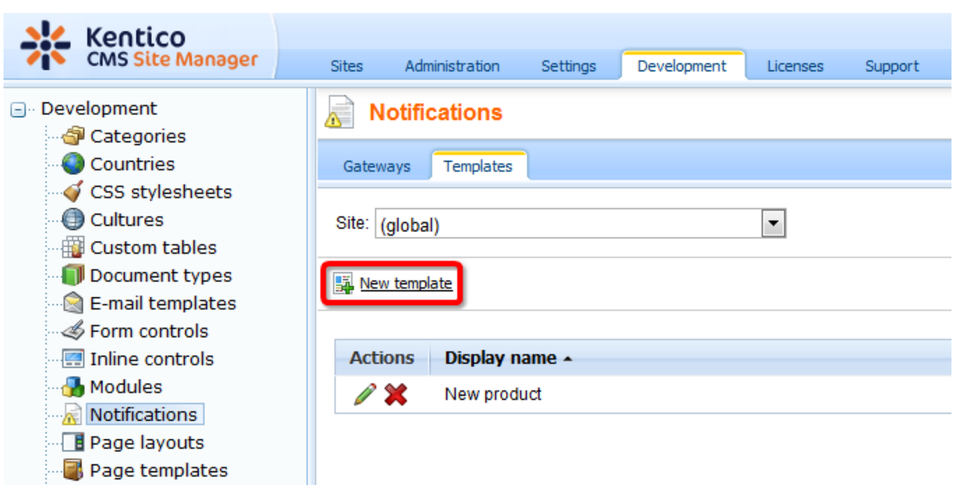
Please note that you will need to have the Corporate Site sample website installed to follow examples in this chapter.

Kentico CMS Community Site Guide contains another example of the use of the Notifications module. You can find an example of signing up for receiving notifications on the occasion of news items publishing on the site in Community Site Guide -> Part 2 -> Pre-development tasks -> Creating the News notifications template.

Please note that you will need to have the Community Site sample website installed to follow Kentico CMS Community Site Guide.

8.29.2 Creating a notification message template

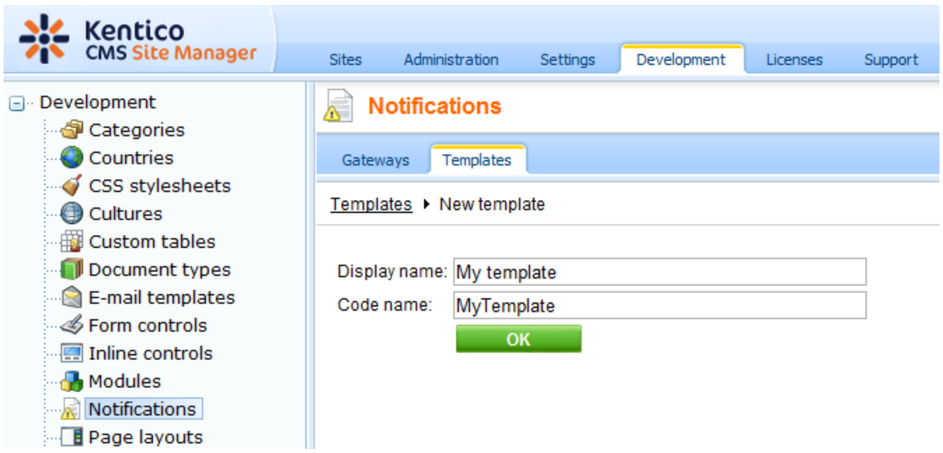
1. Go to **Site Manager -> Development -> Notifications** and switch to the **Templates** tab. Using the **Sites** drop-down list, you can select which site will be the template used by. If you choose **(global)**, the template will be available for all sites. When selected, click the **New template** link. You will be redirected to the **New template** form.



2. On the **New template** form, enter the following details:

- **Display name** - display name of the template
- **Code name** - code name of the template

Click **OK**. The template will be created.



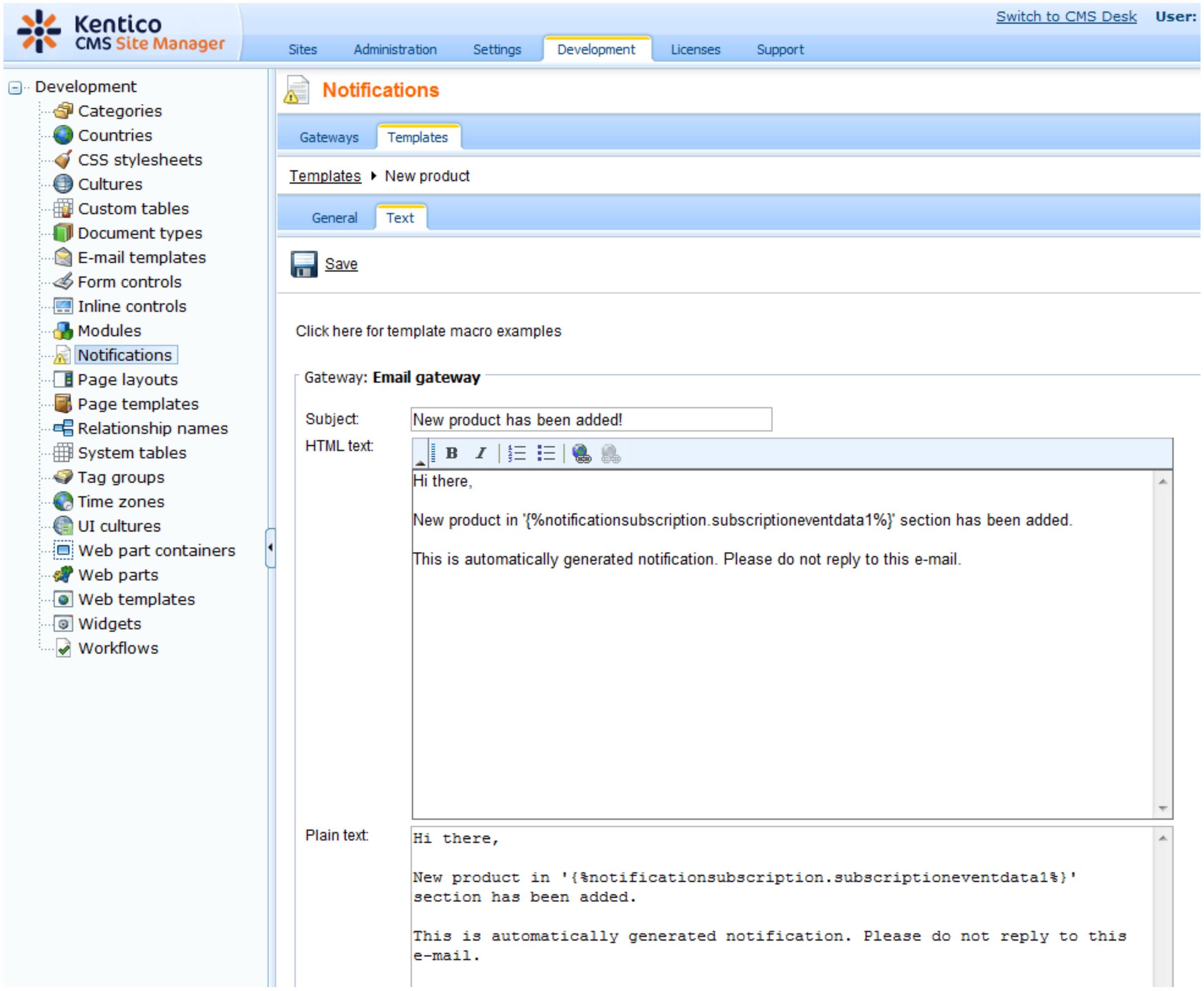
3. Now you should see the same form that you've just filled, but in the **General** tab, while another tab called **Text** is also available. Switch to the **Text** tab.

On this tab, the text of the notification message can be defined. You should see **sections for each of the defined gateways**. Texts can be set separately for each gateway, so that for one event, you can have different texts sent via e-mail and via SMS. Each of the sections contains the gateway name and some of the following three fields, depending on the gateway settings:

- **Subject** - message subject text
- **HTML text** - text of the message in HTML format
- **Plain text** - text of the message in plain text format

The following macros can be used in your notification templates:

- **{%notificationsubscription.SubscriptionID%}** - displays the value of specified data column from the *Notification_Subscription* table; this represents subscription information
- **{%notificationgateway.GatewayID%}** - displays the value of the specified data column from the *Notification_Gateway* table; this represents the notification gateway which performs the sending of the notification message
- **{%notificationuser.UserID%}** - displays the value of the specified user data column from the *CMS_Usertable* table; this represents the user the notification message is being sent to
- **{%notificationcustomdata.XXX%}** - displays the value of the specified data column from a custom data source. Columns from *View_CMS_Tree_Joined* and the document type's table can be used (e.g. *CONTENT_article* for *CMS.article* document type).
- **{%documentlink%}** - displays the link to the document (for content event notifications only)

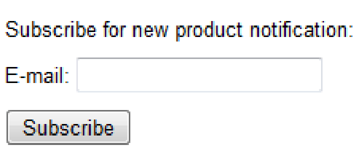


When you have entered appropriate text for each available gateway, click **Save**.

8.29.3 Subscribing users to content changes notifications

The **Content subscription** web part is a pre-configured version of the **Notification subscription** web part. It can be conveniently used to let site visitors subscribe to notifications about the following three events:

- **Document has been created**
- **Document has been updated**
- **Document has been deleted**



In the following example, you will learn how to add the **Content subscription** web part to your site and set it up. We will use the Corporate site sample website as the starting point. Let's presume that we want to enable site visitors to receive an e-mail whenever a new product is added.

Please note

In order for you to see the full functionality, it is necessary to have a SMTP server configured correctly. For more information on how to do this, please refer to the SMTP server configuration chapter.

1. Sign in to **CMS Desk** as the global administrator (login *administrator* with blank password by default). From the content tree on the left, select the **Products** title page and switch to the **Design** tab.

2. Add the **Content subscription** web part to the **zoneLeft** web part zone, below the **Left tree menu**. In the **Web part properties** dialog, set the following properties:

- **Site name** - (current site)
- **Path** - /Products/%
- **Document type** - CMS.Product
- **Event description** - Enter your e-mail address to receive notifications about new products:

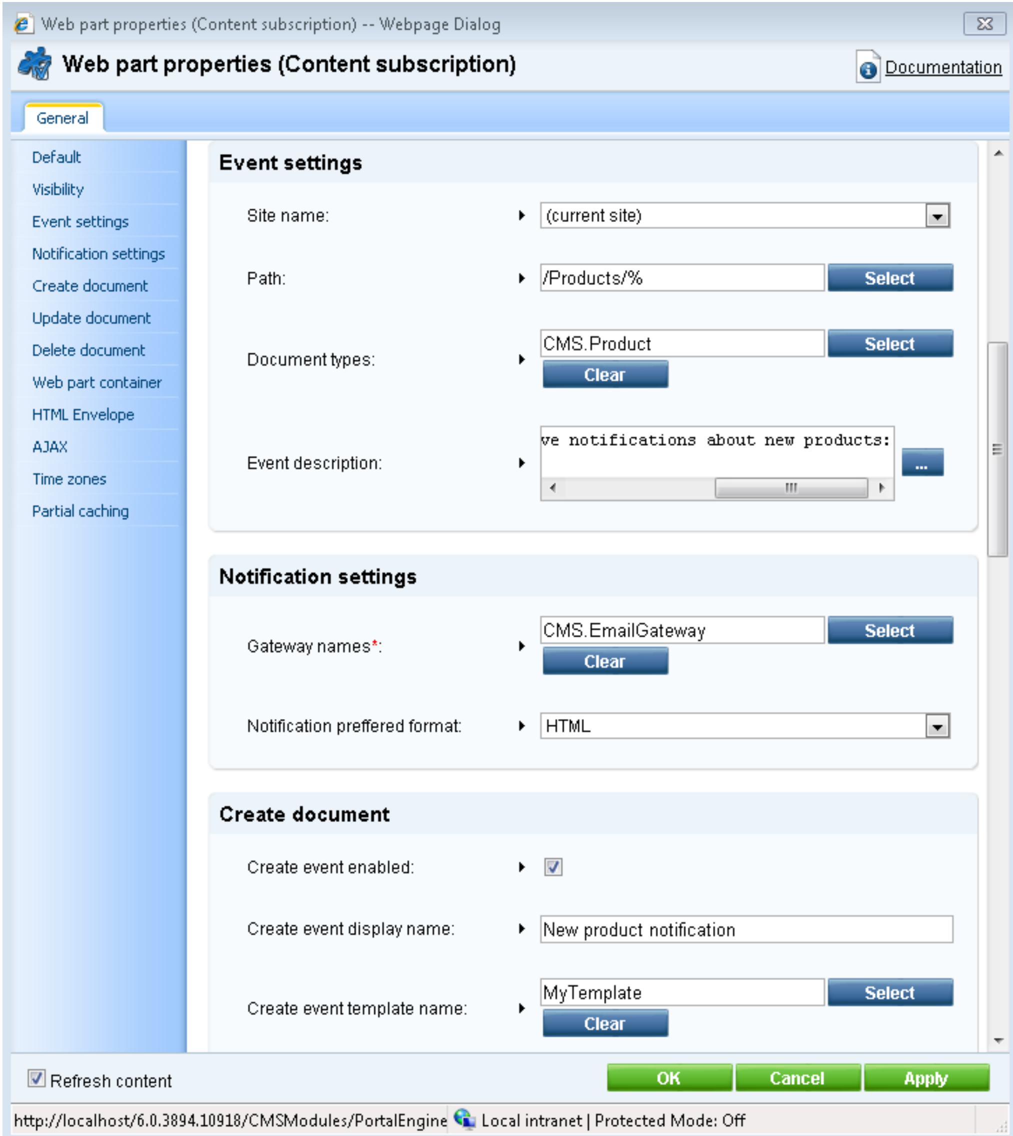
- **Gateway names** - CMS.EmailGateway
- **Notification preferred format** - HTML

- **Create event enabled -** true
- **Create event display name** - New product notification
- **Create event template name** - MyTemplate

and click **OK**.

Please note

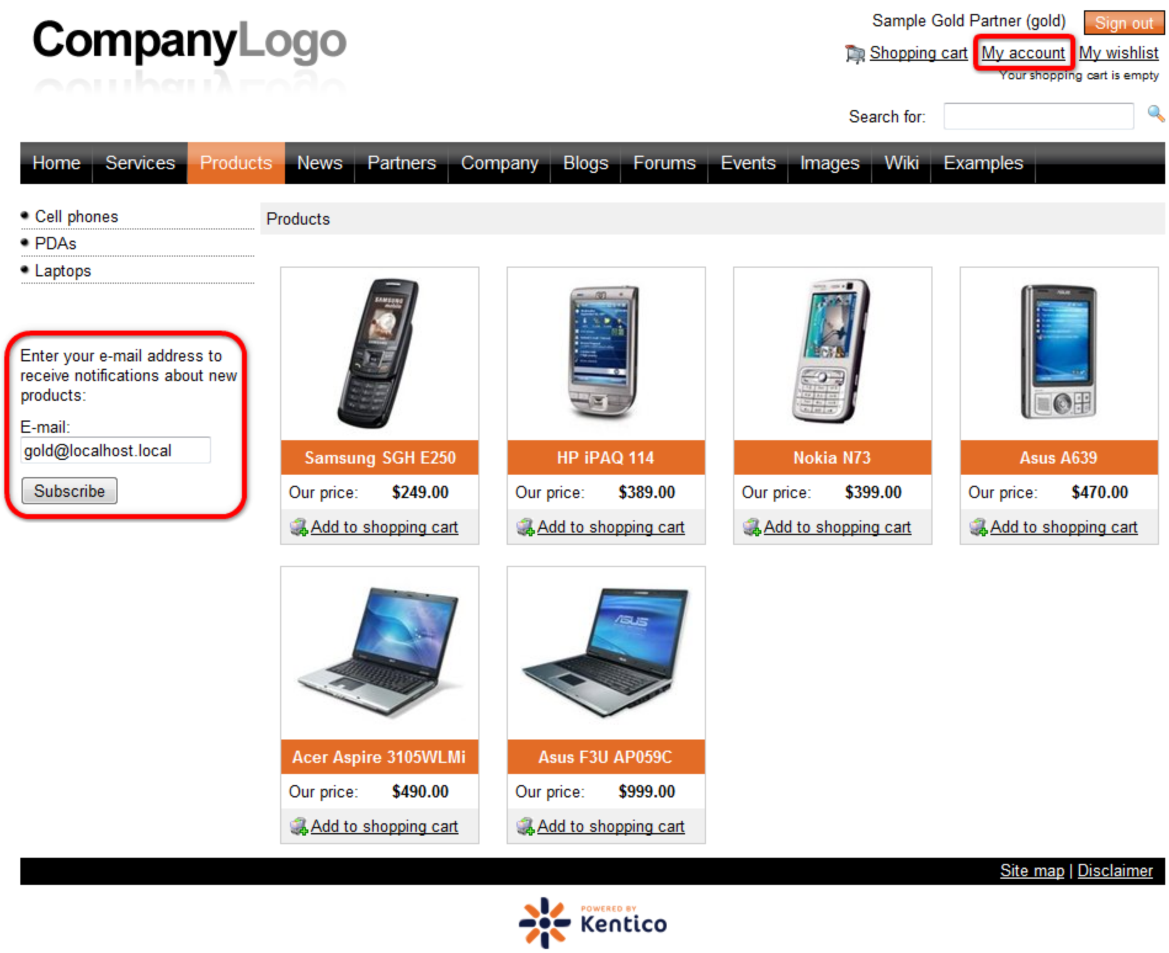
It is important that the template you have recently created according to the *Creating a notification message template* topic is global or assigned to the current site.



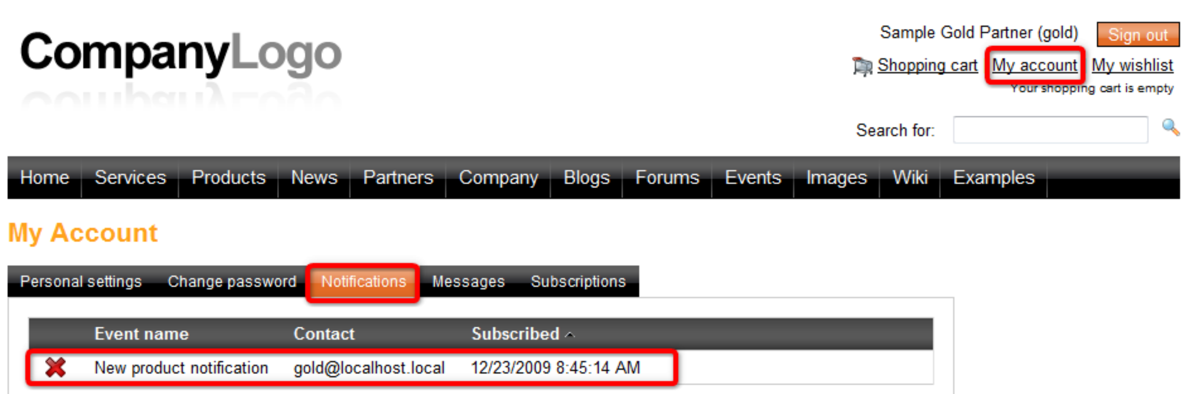
3. Let's switch to the **site visitor's** perspective now. Sign out of **CMS Desk**.

As notifications are available only for registered users, the web part is not visible unless you log in. Use the **My account** link at the top of the page to display a logon form and sign in; use e.g. user name *Gold* with blank password. Once signed in, switch to the **Products** page. You should see the **Content subscription** web part underneath the tree menu as highlighted in the screenshot below.

If you want to receive notifications about new products on this page, you can just type in your e-mail address into the **E-mail** field and click **Subscribe**. In case that you want to verify the functionality later on, enter your own e-mail address so that you can check your inbox for notification e-mails.



4. You can immediately verify your subscription by going to the **My account** section again. If you switch to the **Notifications** section of the **My account** web part, you should see the notification subscription present.

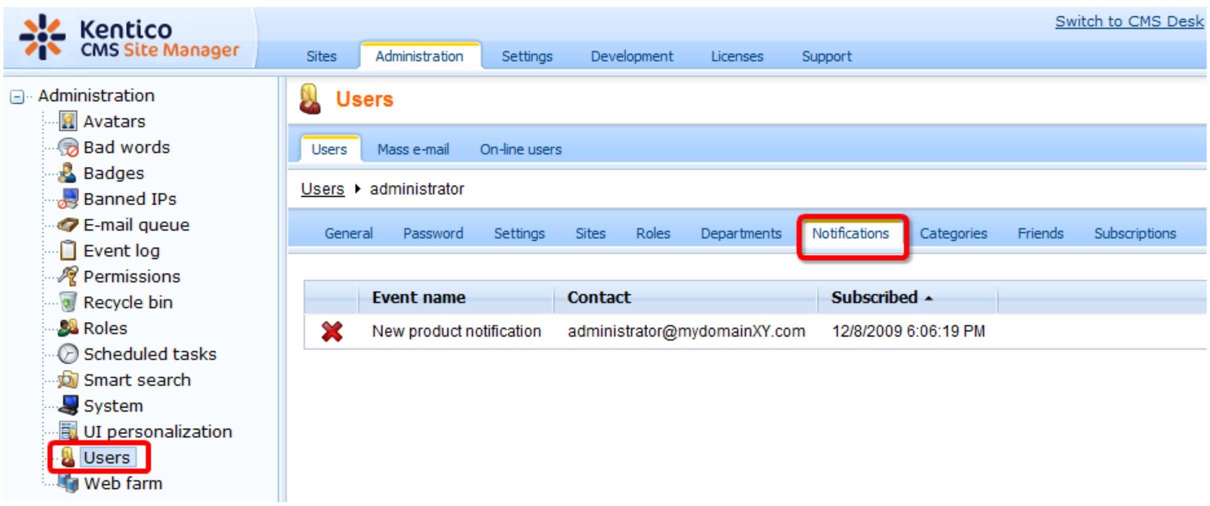


5. Now you can verify that the whole setup works. Log in to **CMS Desk** as the *administrator* again. From the content tree, select **Products** and click the **New** icon above the content tree. Choose to create some new product, e.g. a Cell phone, enter some sample data about the cell phone and click **Save**. Check your e-mail inbox in a few minutes. You should have received a new notification message about

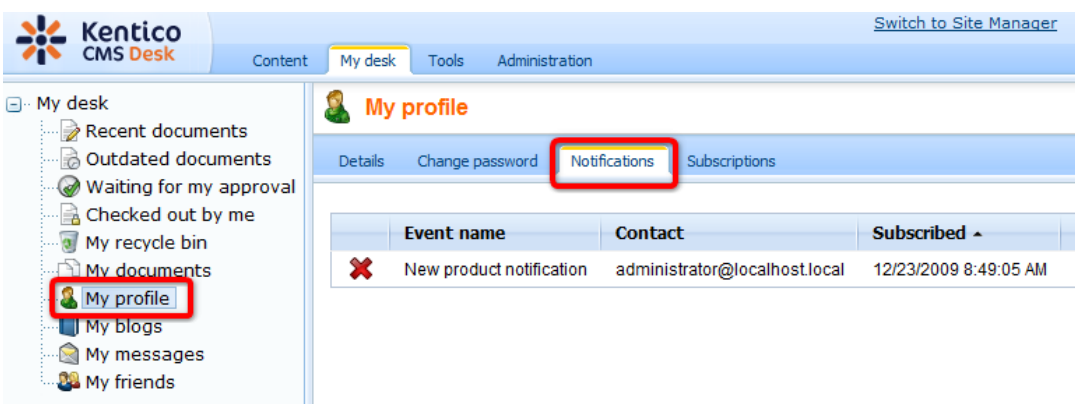
the newly added product.

8.29.4 Managing users' notifications

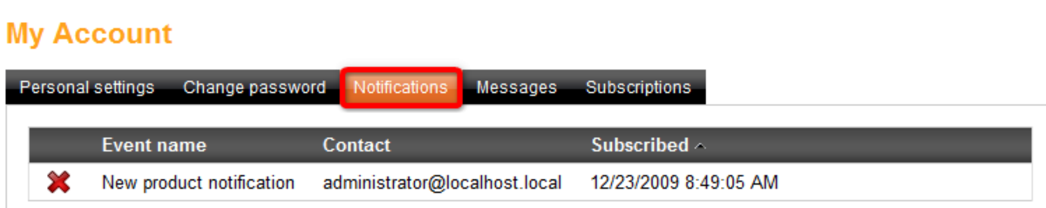
Site administrators can manage subscriptions of particular users in **Site Manager -> Administrator -> Users**. If you choose to **Edit** ( ) a user and switch to the **Notifications** tab, you will see all notifications that the current user is subscribed to. You can unsubscribe the user from a notification by clicking the **Delete** ( ) icon next to it.



CMS Desk users can manage their own subscriptions the same way in **CMS Desk -> My Desk -> My profile -> Notifications**.



Site visitors registered to the site can manage their subscriptions the same way using the **My account** web part, on its **Notifications** tab.



8.29.5 Custom notification gateway

8.29.5.1 Overview

In this chapter, you will get familiar with the process of creating a custom notification gateway and using it on your site. The general description is supplied by code examples of a sample e-mail notification gateway.

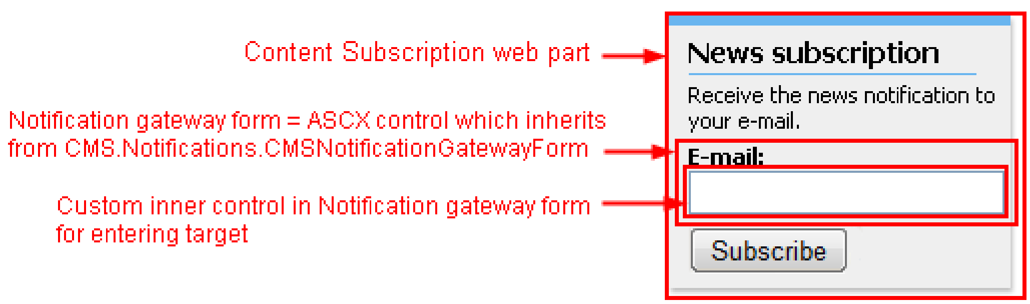
General steps

Here is a general overview of the steps that need to be taken when implementing a custom notification gateway:

1. Create a custom notification gateway form with your custom inner control(s) for entering the target where the notification messages should be sent.
2. Create a custom notification gateway class which handles loading of correct notification gateway form and sending notification messages.
3. Register the custom notification gateway in **Site Manager -> Development -> Notifications** and create a template for notification messages.
4. Use the gateway with the **Content subscription** web part on the live site.

8.29.5.2 Custom notification gateway form

In the picture below, you can see how the default **E-mail notification gateway form** is contained within the **Content subscription** web part.



The following steps need to be taken to create a custom notification gateway form:

1. Create a new web user control (*.ascx) and place it into your site folder which is located in the root of your web project. You can place it anywhere in your web project, but since the control is located in the site folder, it will be included in your site's export packages.
2. Set the control's class to inherit from the ***CMS.Notifications.CMSNotificationGatewayForm*** class.
3. There are two methods and one property you will need to override to reach the required functionality:
   - **object Value** - gets or sets the value from the custom inner control (e.g. gets or sets the text of the inner TextBox in the picture above)
   - **string Validate()** – validates the inner control's value and returns an error message if the value doesn't meet the requirements of the notification gateway (e.g. the TextBox value is validated for e-mail address format for the **E-mail notification gateway**)
   - **void ClearForm()** – your custom code inside this method should set the inner control to the default state; example: text of the TextBox should be cleared (set to the empty string)

Example - E-mail notification gateway form

The following code samples show how a custom e-mail notification gateway form can be implemented:

EmailNotificationForm.ascx

Please note: If you installed the Kentico CMS project as a web application, you need to rename the *CodeFile* property on the first line to *Codebehind* for the code example to be functional.

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="EmailNotificationForm.asc
    Inherits="CMSModules_Notifications_Controls_NotificationSubscription_EmailNotifi
<asp:Label ID="lblEmail" runat="server" />
<asp:TextBox ID="txtEmail" runat="server" CssClass="EmailNotificationForm" EnableVie
```

EmailNotificationForm.ascx.cs

[C#]

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

using CMS.Notifications;
using CMS.GlobalHelper;
using CMS.CMSHelper;
using CMS.EventLog;

public partial class
CMSModules_Notifications_Controls_NotificationSubscription_EmailNotificationForm :
CMSNotificationGatewayForm
{
    #region "Variables"

    private bool mEnableMultipleEmails = false;

    #endregion


    #region "Properties"

    /// <summary>
    /// Gets or sets the e-mail(s) from/to textbox.
    /// </summary>
    public override object Value
```

```
    {
        get
        {
            return this.txtEmail.Text.Trim();
        }
        set
        {
            this.txtEmail.Text = ValidationHelper.GetString(value, "");
        }
    }


    /// <summary>
    /// Gets or sets the value which determines whether to allow multiple e-mails
separated with semicolon.
    /// </summary>
    public bool EnableMultipleEmails
    {
        get
        {
            return this.mEnableMultipleEmails;
        }
        set
        {
            this.mEnableMultipleEmails = value;
        }
    }

    #endregion


    protected void Page_Load(object sender, EventArgs e)
    {
        this.lblEmail.Text = (this.EnableMultipleEmails ? ResHelper.GetString(
"general.emails") : ResHelper.GetString("general.email")) + ResHelper.Colon;

        // Fill in the default e-mail
        if ((!RequestHelper.IsPostBack()) && (CMSContext.CurrentUser != null) &&
(!String.IsNullOrEmpty(CMSContext.CurrentUser.Email)))
        {
            this.txtEmail.Text = CMSContext.CurrentUser.Email;
        }
    }


    #region "Public methods"

    /// <summary>
    /// Checks whether the input is correct e-mail address (or multiple e-mail
addresses).
    /// </summary>
    public override string Validate()
    {
        if (this.EnableMultipleEmails)
```

```
        {
            if (!ValidationHelper.AreEmails(this.txtEmail.Text.Trim()))
            {
                return ResHelper.GetString("notifications.emailgateway.formats");
            }
        }
        else
        {
            if (!ValidationHelper.IsEmail(this.txtEmail.Text.Trim()))
            {
                return ResHelper.GetString("notifications.emailgateway.format");
            }
        }

        return String.Empty;
    }


    /// <summary>
    /// Clears the e-mail textbox field.
    /// </summary>
    public override void ClearForm()
    {
        this.txtEmail.Text = "";
    }

    #endregion
}
```

8.29.5.3 Custom notification gateway class

The following steps need to be taken to create a custom notification gateway class:

1. Create a new library (assembly) as a part of your solution and a new class inside this library.
2. Set your class to inherit from the **CMS.Notifications.CMSNotificationGateway** abstract class.
3. There are two methods you will need to override to reach the required functionality:

- **void SendNotification()** – sends a single notification; it is automatically called after the specified event is raised
- **CMSNotificationGatewayForm GetNotificationGatewayForm()** – loads and returns notification gateway form for the notification gateway

4. Compile the library.
5. Ensure the library file (*.dll) is included in the **/Bin** directory.

Example - E-mail notification gateway class

The following code sample shows how a custom e-mail notification gateway class can be implemented:

EmailNotificationGateway.cs

[C#]

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Web.UI;

using CMS.EventLog;
using CMS.EmailEngine;
using CMS.GlobalHelper;
using CMS.SiteProvider;
using CMS.SettingsProvider;
using CMS.Notifications;

namespace EmailNotification
{
    /// <summary>
    /// Base class for e-mail notification gateway.
    /// </summary>
    public class EmailNotificationGateway : CMSNotificationGateway
    {
        /// <summary>
        /// Returns the e-mail gateway form.
        /// </summary>
        public override CMSNotificationGatewayForm GetNotificationGatewayForm()
        {
            try
            {
                Control ctrl = this.NotificationSubscriptionControl.Page.
LoadControl("~/CMSModules/Notifications/Controls/NotificationSubscription/
EmailNotificationForm.ascx");
                ctrl.ID = ValidationHelper.GetIdentificator("notif" + this.
NotificationGatewayObj.GatewayName);

                return (CMSNotificationGatewayForm)ctrl;
            }
            catch (Exception ex)
            {
                try
                {
                    // Log the event
                    EventLogProvider eventLog = new EventLogProvider();
                    eventLog.LogEvent("EmailGateway", "EXCEPTION", ex);
                }
                catch
                {
                    // Unable to log the event
                }
            }
            return null;
        }


        /// <summary>
        /// Sends the notification via e-mail.
```

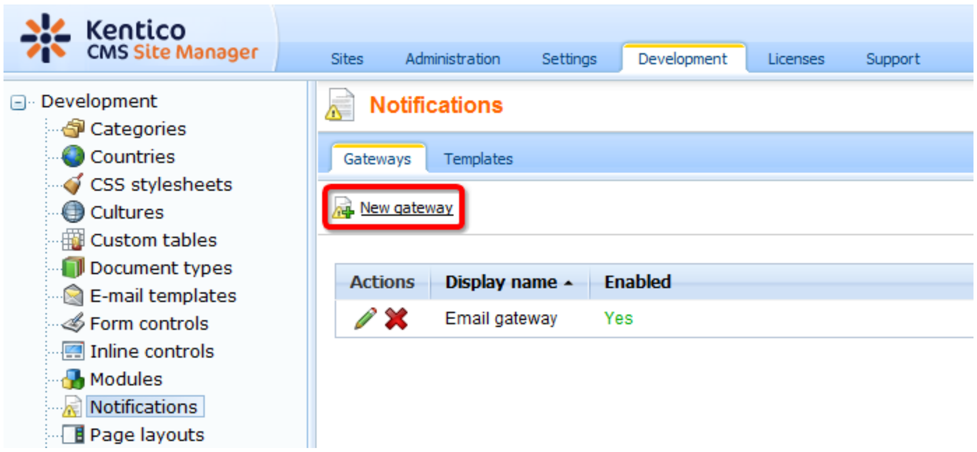
```
        /// </summary>
        public override void SendNotification()
        {
            try
            {
                if (this.NotificationSubscriptionObj != null)
                {
                    // Get template text
                    NotificationTemplateTextInfo templateText =
NotificationTemplateTextInfoProvider.GetNotificationTemplateTextInfo(this.
NotificationGatewayObj.GatewayID, this.NotificationSubscriptionObj.
SubscriptionTemplateID);
                    if (templateText != null)
                    {
                        // Get the site name
                        string siteName = null;
                        SiteInfo si = SiteInfoProvider.GetSiteInfo(this.
NotificationSubscriptionObj.SubscriptionSiteID);
                        if (si != null)
                        {
                            siteName = si.SiteName;
                        }

                        // Create message object
                        EmailMessage message = new EmailMessage();

                        // Get sender from settings
                        message.From = SettingsKeyProvider.GetStringValue(siteName
+ ".CMSSendEmailNotificationsFrom");

                        // Do not send the e-mail if there is no sender specified
                        if (message.From != "")
                        {
                            // Initialize message
                            message.Recipients = this.NotificationSubscriptionObj.
SubscriptionTarget;

                            // Body
                            if ((this.NotificationSubscriptionObj.
SubscriptionUseHTML) &&
                                (this.NotificationGatewayObj.
GatewaySupportsHTMLText) &&
                                (templateText.TemplateHTMLText != ""))
                            {
                                // HTML format, set Body property, resolve macros
if possible
                                message.EmailFormat = EmailFormatEnum.Html;
                                if (this.Resolver != null)
                                {
                                    message.Body = this.Resolver.ResolveMacros
(templateText.TemplateHTMLText);
                                }
                                else
                                {
```

```
                                    message.Body = templateText.TemplateHTMLText;
                                }
                            }
                            else
                            {
                                // Plaintext format, set PlainTextBody property,
resolve macros if possible
                                message.EmailFormat = EmailFormatEnum.PlainText;
                                if (this.Resolver != null)
                                {
                                    message.PlainTextBody = this.Resolver.
ResolveMacros(templateText.TemplatePlainText);
                                }
                                else
                                {
                                    message.PlainTextBody = templateText.
TemplatePlainText;
                                }
                            }

                            // Subject, resolve macros if possible
                            if (this.Resolver != null)
                            {
                                message.Subject = this.Resolver.ResolveMacros
(templateText.TemplateSubject);
                            }
                            else
                            {
                                message.Subject = templateText.TemplateSubject;
                            }

                            // Send email via Email engine API
                            EmailSender.SendEmail(siteName, message);
                        }
                    }
                }
            }
            catch (Exception ex)
            {
                try
                {
                    // Log the event
                    EventLogProvider eventLog = new EventLogProvider();
                    eventLog.LogEvent("EmailGateway", "EXCEPTION", ex);
                }
                catch
                {
                    // Unable to log the event
                }
            }
        }
    }
}
```

8.29.5.4 Registering a custom gateway

When you have developed a custom notification gateway, you need to take the following steps to register your gateway in the system:

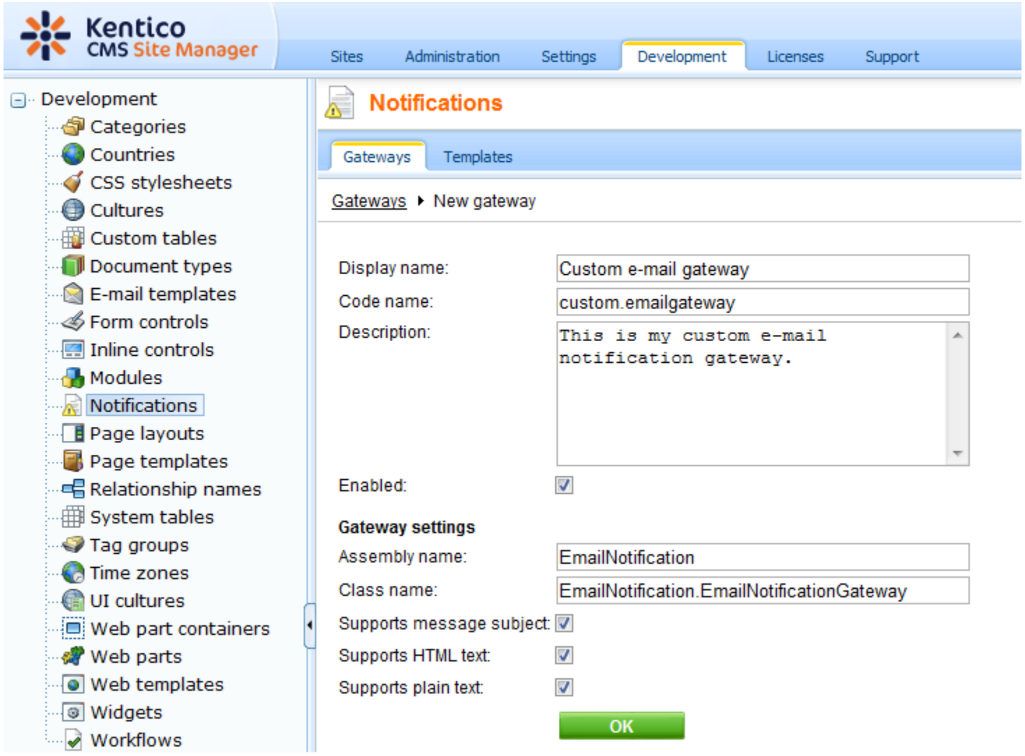
1. Go to **Site Manager -> Development -> Notifications**. On the **Gateways** tab, you can find the **New gateway** link at the top of the page. Click it, the **New gateway form** will be displayed.



2. Enter the following details into the **New gateway** form:

- **Display name** - display name of the notification gateway
- **Code name** - code name of the gateway
- **Description** - text describing the gateway
- **Enabled** - if unchecked, the notification gateway is not functional - this can be useful if you want to temporarily disable the gateway so that no messages will be sent, e.g. when you are performing some administration tasks; if checked, the gateway works normally

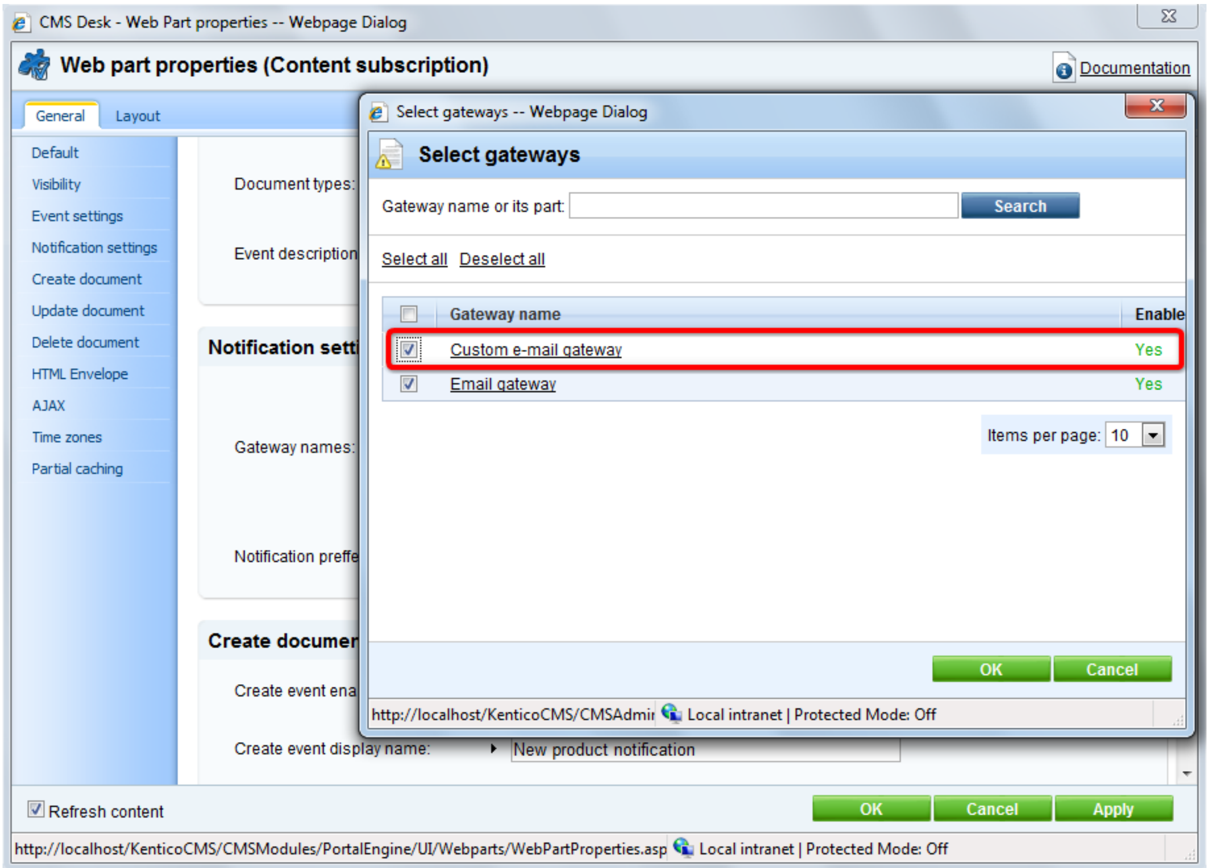
- **Assembly name** - name of the assembly in that the gateway code is stored
- **Class name** - name of the class containing the gateway code; must be entered including the assembly name, as you can see in the screenshot below
- **Supports message subject** - enable if the gateway's message format supports message subjects
- **Supports HTML text** - enable if the gateway supports messages in HTML format (e.g. for e-mails)
- **Supports plain text** - enable if the gateway supports plain text format (e.g. for SMS)



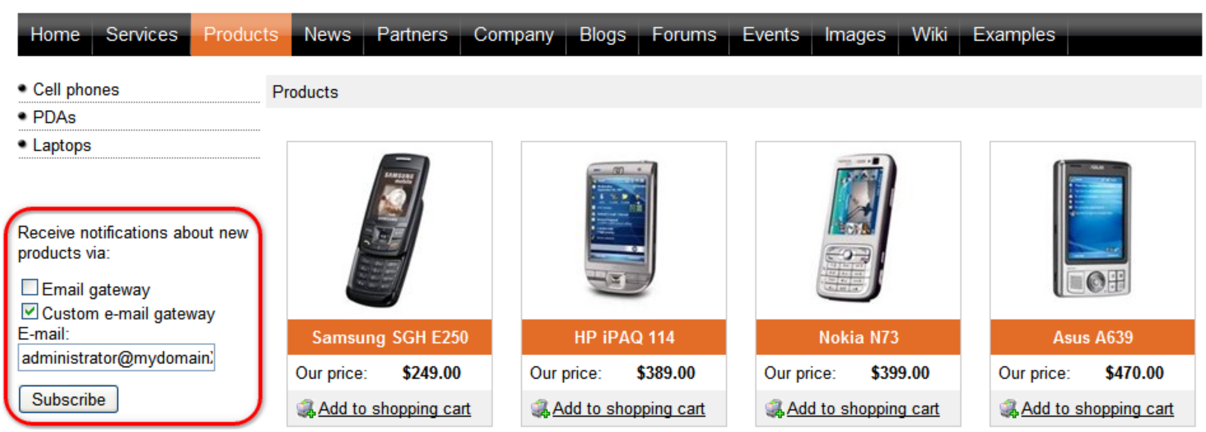
3. After entering all the details, click **OK** to confirm the changes you have made.

8.29.5.5 Using the gateway on your site

On the live site, you can enable users to subscribe using the **Content subscription** web part, just as described here. You only need to enable your gateway in the **Gateway names** property of the web part.



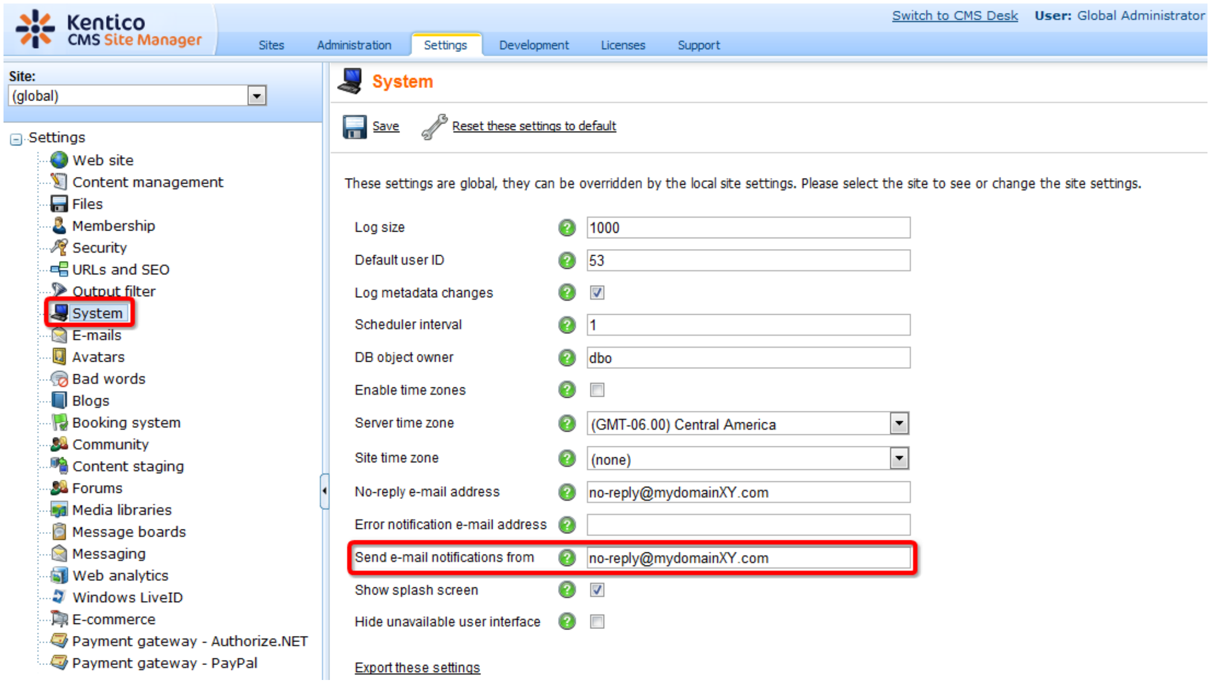
On the live site, the web part displays a check-box for each gateway when multiple gateways are selected. If you enable a particular check-box, the gateway's form is displayed, as you can see in the screenshot below.



8.29.6 Settings

The **Notifications** module has only one setting to be done in **Site Manager -> Settings -> System**:

- **Send e-mail notifications from** - sets the e-mail address that will be used as the sender address (the **From** field) of notification e-mails



8.29.7 Security

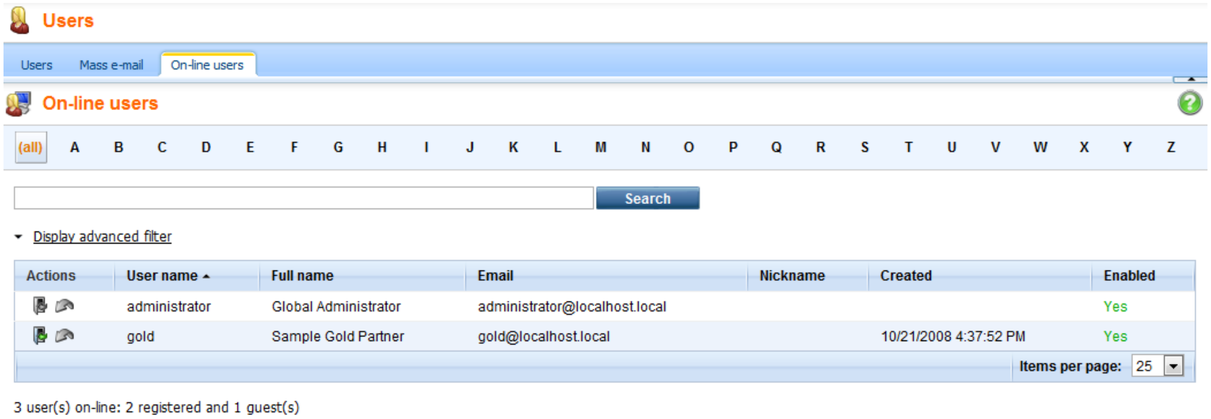
The **Notifications** module has no permissions to be set in **CMS Desk/Site Manager -> Administration -> Permissions**.

Subscription to notifications is **allowed only for registered users**. This is why the **Content subscription** and **Notification subscription** web parts are **hidden to public anonymous users** by default.

8.30 On-line users

8.30.1 Overview

The On-line users module allows you to monitor users currently connected to the website. This can be useful for various site administration purposes. In addition to providing information, the module also provides a way to temporarily kick a specific user off the website.



The module identifies a new user when a new session between the client browser and the server is started. The user is considered off-line if the session expires or when the user logs off. This means that a user is still considered on-line for some time when they close their web browser without signing off.

To utilize the module on your site, you need to perform the following simple steps:

1. Enable and configure on-line user monitoring in the system. Learn how in the Enabling the On-line users module topic.

2. Keep track of users on the On-line users tab or display information about them on your website using the On-line users web part.

The On-line users internals and API sub-chapter provides information about the database tables and classes used by the module and examples of how on-line users can be managed using the API.

For more general information about users and site membership, please refer to the Development -> Membership, permissions and security chapter of this guide.

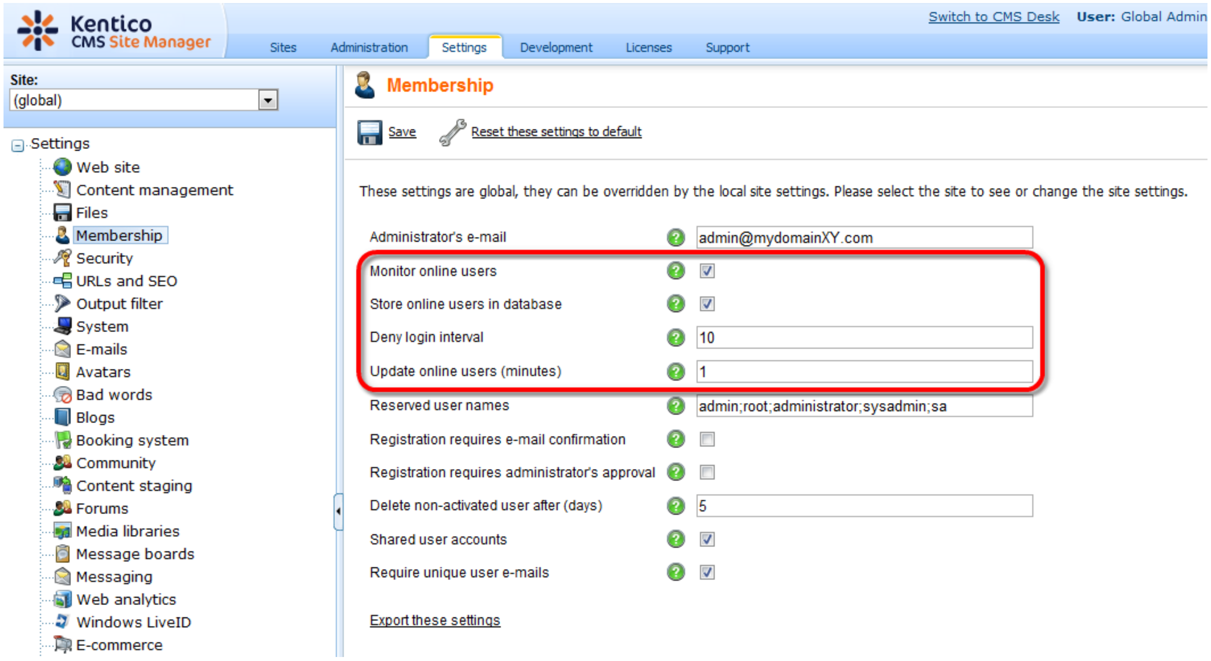
8.30.2 Enabling the On-line users module

The On-line users module can be enabled in **Site Manager -> Settings -> Membership** by checking the **Monitor online users** check-box.

If you are running the system on a web farm, you also have to check the **Store online users in database** check-box. This causes information about on-line users to be saved in the database.

The **Deny login interval** property determines how long users will not be able to log-in after they are kicked. The value is entered in minutes.

Finally, the **Update online users (minutes)** property defines how often information about users accessing the site is reloaded. The value is entered in minutes. When running the system on a web farm, you need to enter the same value which is set for the **Remove expired sessions** scheduled task (you can read the value in **Site Manager -> Administration -> Scheduled tasks -> edit () Remove expired sessions -> Task interval -> Every: X minute**).



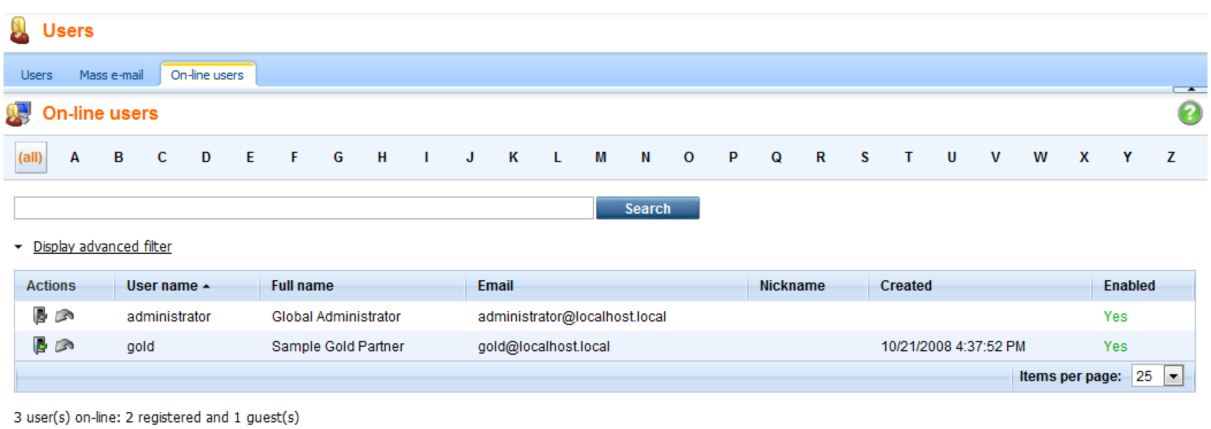
8.30.3 On-line users tab

If you go to **Site Manager/CMS Desk -> Administration -> Users** and switch to the **On-line users** tab, you can see a list of all users that are currently accessing the website. By clicking some of the letters at the top of the page, you can view only those users whose user names begin with the clicked letter.

Below the letters, you can find a filter. This filter can further limit displayed users according to the specified criteria. The filter can work in two modes between which you can switch by clicking the **Display advanced filter** or **Display simplified filter** links respectively.

Simplified filter

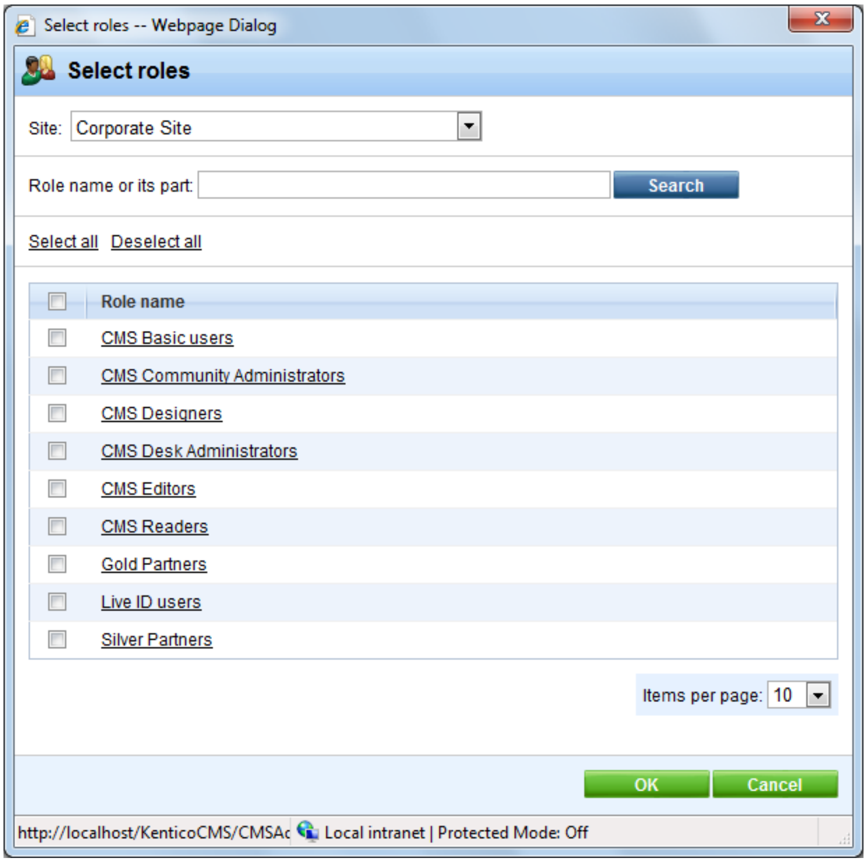
This option offers only one field. The expression entered into the field will be searched in users' **User Name**, **Full Name**, **E-mail Address** and **Nickname** properties.



Advanced filter

The advanced filter offers searching by **User Name**, **Full Name**, **E-mail Address** and **Nickname**.

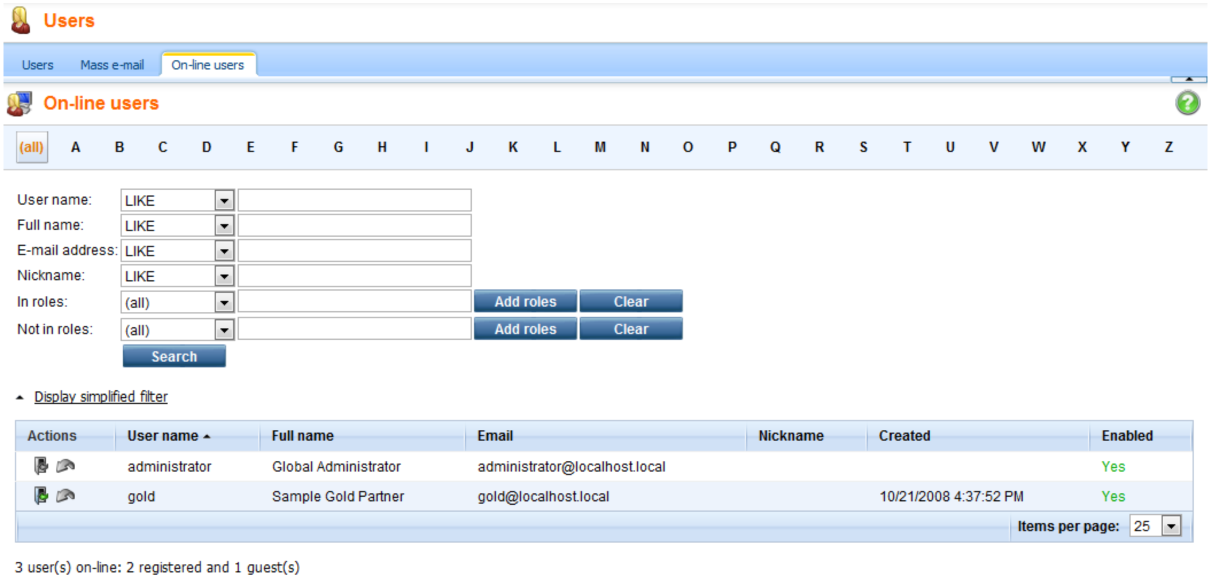
You can also filter on-line users by the roles they belong to. You can specify which roles the listed users are (**In roles**) or are not (**Not in roles**) members of by using the **Add roles** button to open the role selection dialog. Click **OK** to confirm the selection.



If you have selected more than one user role, you can also select if the displayed users should (not) be members of **All** or **Any** of the selected roles.

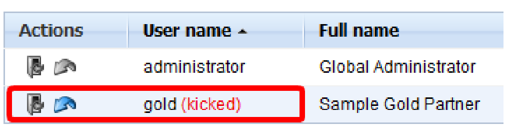
When you have entered all search criteria, click the **Search** button.

Only those users that match the specified criteria will be listed.



Kicking a user

If you click the **Kick** ( ) icon next to one of the listed users, you can kick the user off of the website, which means that the user will be logged from of the website and won't be able to log in for the duration specified in the **Site Manager -> Settings -> Membership -> Deny login interval** setting. After doing so, there will be a label (kicked) written in red letters after the user's user name. If you want to take back the kick, you can do so easily by clicking the **Take back** ( ) icon.



If the reason for kicking a user is serious enough, you may want to consider using the Banned IPs module to permanently block the user from accessing the website.

8.30.4 On-line users web part

The module comes with the **On-line users** web part. In the web part selection dialog, you can find the web part in the **Membership** category. In the screenshot below, you can see the web part enclosed in the **Orange box** web part container.

The web part displays a summary defined by the **Additional info text** property, followed by a list of on-line users that is displayed based on the transformation defined by the **Transformation name** property.

The web part has the following related properties:

Transformation name	Name of the transformation that will be used to display on-line users.

	If you choose to **Select** the transformation, you can find the default transformation that has been used in the screenshot above at **Site Manager -> Development -> Document types -> CMS.Root -> Transformations -> CMS.Root.OnLineUsers**.
Path	If you specify an alias path here, only users that are accessing locations found under the specified path will be displayed.
Select top N	Number of listed users.
Additional info text	Text of the additional information that will be displayed above the list of users. You can use the following formatting macros that will be substituted by the appropriate number: {0} - number of all connected users {2} - number of connected registered users {1} - number of connected anonymous users
No users on-line text	Text that will be displayed if no users are currently online.
Columns	Enumeration of columns of the **CMS_User** or **CMS_UserSettings** tables that should be loaded - useful for performance optimization. Column names need to be separated by commas (,).

8.30.5 On-line users internals and API

8.30.5.1 Database tables and API classes

The On-line users modules uses the following database table:

- **CMS_Session** - stores information about on-line users when the **Site Manager -> Settings -> Membership -> Store on-line users in database** settings is enabled; this is necessary when running the system on a web farm

The On-line users API is provided by the following **CMS.CMSHelper** namespace class:

- **SessionManager** - this class provides functionality for managing sessions, which includes the monitoring of on-line users

The following topics show examples of how this class can be used:

- Getting on-line user data
- Checking if a user is on-line
- Kicking a user

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.30.5.2 Getting on-line user data

The following sample code shows how you can get a DataSet containing on-line users filtered by various parameters:

[C#]

```
using System.Data;
using CMS.CMSHelper;

...

        string where = "UserID > 53";
        int topN = 10;
        string orderBy = "UserName";
        string columns = "UserName, Email";

        // Only users accessing the specified location (alias path) will be selected
        string location = "/Home";

        string siteName = "CorporateSite";
        bool includeHidden = true;
        bool includeKicked = false;

        // Get DataSet of online users according to the given parameters
        DataSet ds = SessionManager.GetOnlineUsers(where, orderBy, topN, location,
siteName, includeHidden, includeKicked);

        DataSet ds2 = SessionManager.GetOnlineUsers(where, orderBy, topN, columns,
location, siteName, includeHidden, includeKicked);
```

8.30.5.3 Checking if a user is on-line

The following sample code shows how you can check if a user specified by their ID is on-line on a given site; the *IsUserOnline(...)* method returns a boolean value:

[C#]

```
using CMS.CMSHelper;
using CMS.SiteProvider;

...

        string siteName = "CorporateSite";
        int userID = 10;
        bool includeHidden = true;

        // Get user and site objects
        UserInfo ui = UserInfoProvider.GetUserInfo(userID);
        SiteInfo si = SiteInfoProvider.GetSiteInfo(siteName);
```

```
        // If both exist
        if ((ui != null) && (si != null))
        {
            // Check if user is online
            SessionManager.IsUserOnline(siteName, userID, includeHidden);
        }
```

8.30.5.4 Kicking a user

The following sample code shows how you can kick a user specified by their ID:

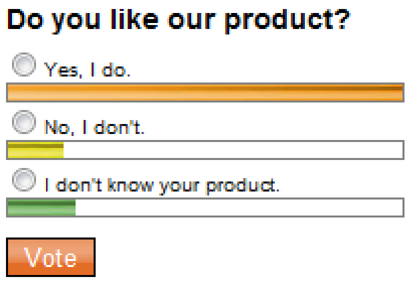
[C#]

```
using CMS.CMSHelper;
using CMS.SiteProvider;

...

        int userID = 10;

        // Get user object
        UserInfo ui = UserInfoProvider.GetUserInfo(userID);

        // Check if user exists
        if (ui != null)
        {
            // Kick user
            SessionManager.KickUser(userID);
        }
```

8.31 Polls

8.31.1 Overview

The Polls module allows you to create, edit and publish polls on the web site created with Kentico CMS. The module can display voting statistics in the form of a graph, which represents either the absolute number of votes or percentage. Besides, the module enables you to control whether single or multiple answers will be allowed on the page and you have the possibility to deny multiple votes for one site visitor.



By placing the Polls module on the pages of your website, you give site visitors the chance to vote for certain options. These options can be related directly to the current page, to the contents of the whole website or simply to anything you would like the visitors of your web pages to express their opinion on. Polls in Kentico CMS are customizable and the module supports integration with the built-in WYSIWYG editor.

The polls can be managed in the **CMS Desk -> Tools -> Polls** dialog. To ensure that the same visitor (using the same browser) cannot vote twice in the same poll, the Polls module uses a cookie.

- To learn how to perform administrative tasks related to polls, please refer to the Managing polls topic.
- To learn how to make your polls available from your web pages, please refer to the Publishing polls topic.
- If you would like to see an example of how to integrate polls on your site, please refer to the Adding a poll to your site topic.
- To learn how to make polls in more than one cultural version, please refer to the Multilingual support topic.
- If you would like to learn about the security possibilities of the Polls module, please refer to the Security topic.

Kentico CMS E-commerce Guide contains another example of the use of the Polls module. You can find a brief description of the module in E-commerce Guide -> Integration with your existing Kentico CMS web site -> Tools -> Polls.

You will need to have the E-commerce Site sample website installed to follow Kentico CMS E-commerce Guide.

A practical example of the use of the Polls module can be found in Kentico CMS Personal Site Guide . You can find a step-by-step tutorial on how to add a poll to your personal website in Personal Site Guide -> Adding web parts -> Adding a poll.

You will need to have the Personal Site sample website installed to follow Kentico CMS Personal Site Guide.

Several more practical examples of the use of the Polls module are available in Kentico CMS Community Site Guide; please note that these examples do not concern the whole functionality of the module but focus on its use in a broader context of the Community Site sample website:

- See Community Site Guide -> Part 1 -> Getting started -> First sight: A general outline of the Community Site and its features; a brief reference to polls is made in paragraph 3.
- See Part 2 -> Pre-development tasks -> Creating a sample poll in Kentico CMS Community Site Guide: You will learn how to create a sample poll that you will publish on the Home page of you website later on.
- See Creating the Home page in the same section of Kentico CMS Community Site Guide: Here you will learn how to publish the previously created sample poll on the Home page your website.

You will need to have the Community Site sample website installed to follow Kentico CMS Community Site Guide.

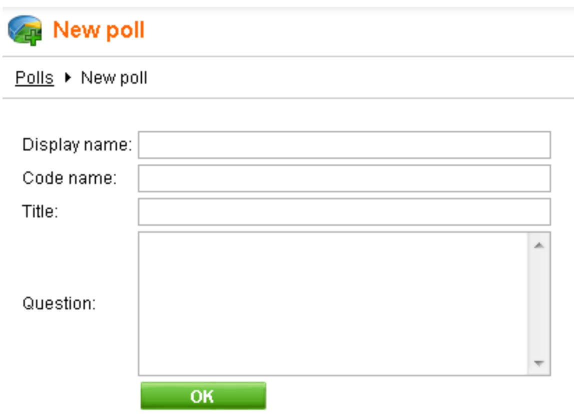
8.31.2 Managing polls

You can manage polls in **CMS Desk -> Tools -> Polls**.

Creating a new poll

Click the **New poll** link and enter the following values:

Display name	The name of the poll displayed to the poll administrators.
Code name	The name of the poll used in the code.
Title	The title of the poll displayed in the poll view (optional).
Question	Poll question displayed in the poll view.



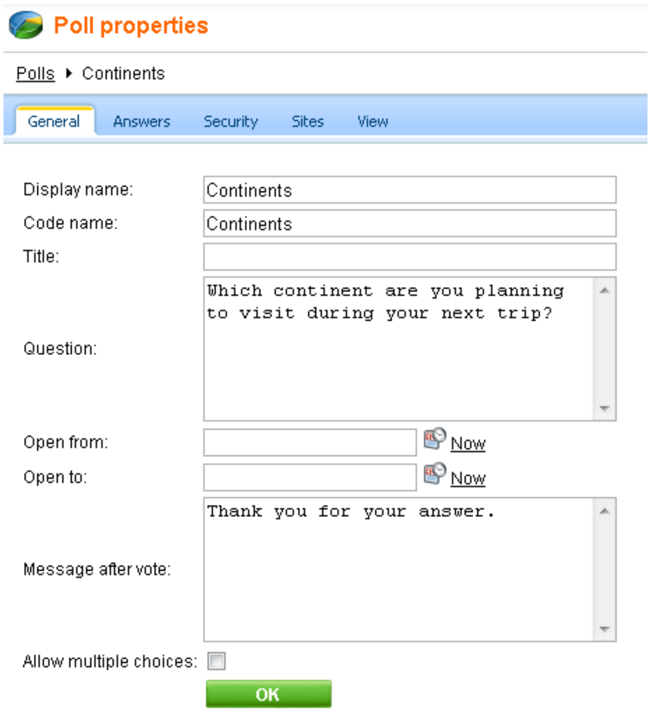
After you have entered the required values, click **OK** to save the changes you have made.

Editing poll properties

By entering the following values, you can now define more poll details on the **General** tab:

Display name	The name of the poll displayed to the poll administrators.
Code name	The name of the poll used in the code.
Title	The title of the poll displayed in the poll view (optional).
Question	Poll question displayed in the poll view.
Open from	When the visitors can start voting.
Open to	When the voting is closed.

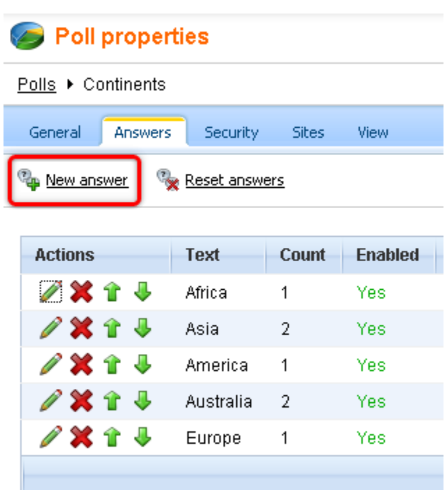
Message after vote	Message displayed after vote.
Allow multiple choices	Indicates if visitor can select more than one option.



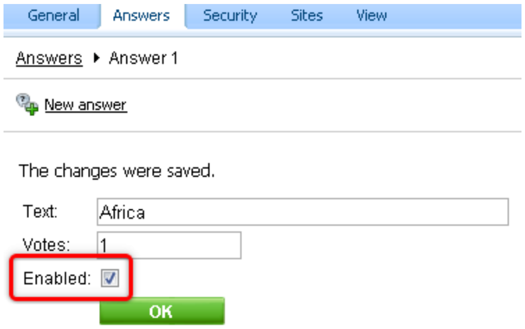
Click **OK** to save the changes you have made to the poll.

Defining answers

Then you can define a list of available answers on the **Answers** tab. Poll answers can be **Edited** (), **Deleted** () or **Moved Up** () and **Moved down** (). To create a new answer, click the **New answer** link and enter the answer text.



You can also choose if the given answer should be enabled, which means it is displayed in the list of options. This is useful if you need to remove an answer from the poll while keeping the number of votes in the history. The disabled option is then not calculated into the displayed results.

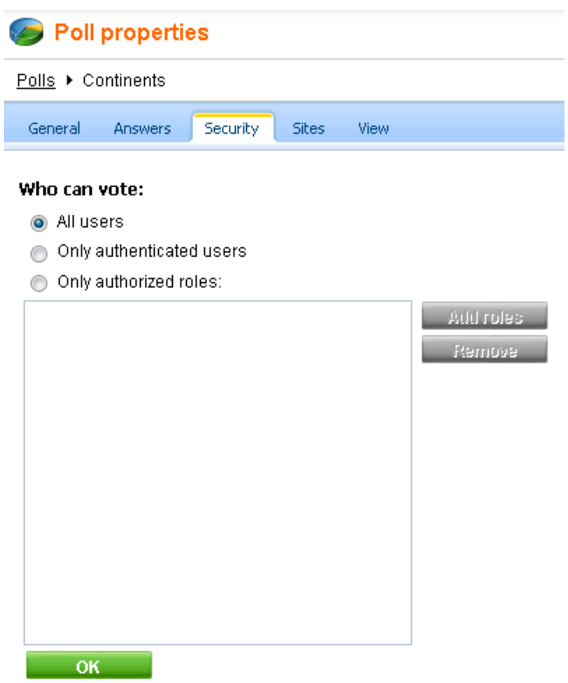


You can also reset answers in your poll by clicking the **Reset answers** link.

Defining who can vote

On the **Security** tab, you can choose which users can vote:

- all users
- only authenticated users (site members)
- only authorized roles that you specify



Sharing the poll between sites

On the **Sites** tab, you can choose on which sites the poll can be used. The poll will be offered to content editors of the websites so that they can put the poll into the text by means of the WYSIWYG editor. By default, the site where you created the poll is added.

Previewing the poll

You can preview the poll on the **View** tab. The actual poll on your website may use a different design depending on the design of your website and may behave differently depending on the web part settings (in case you publish it using a web part).

8.31.3 Publishing polls

There are two ways how you can publish a poll on the website:

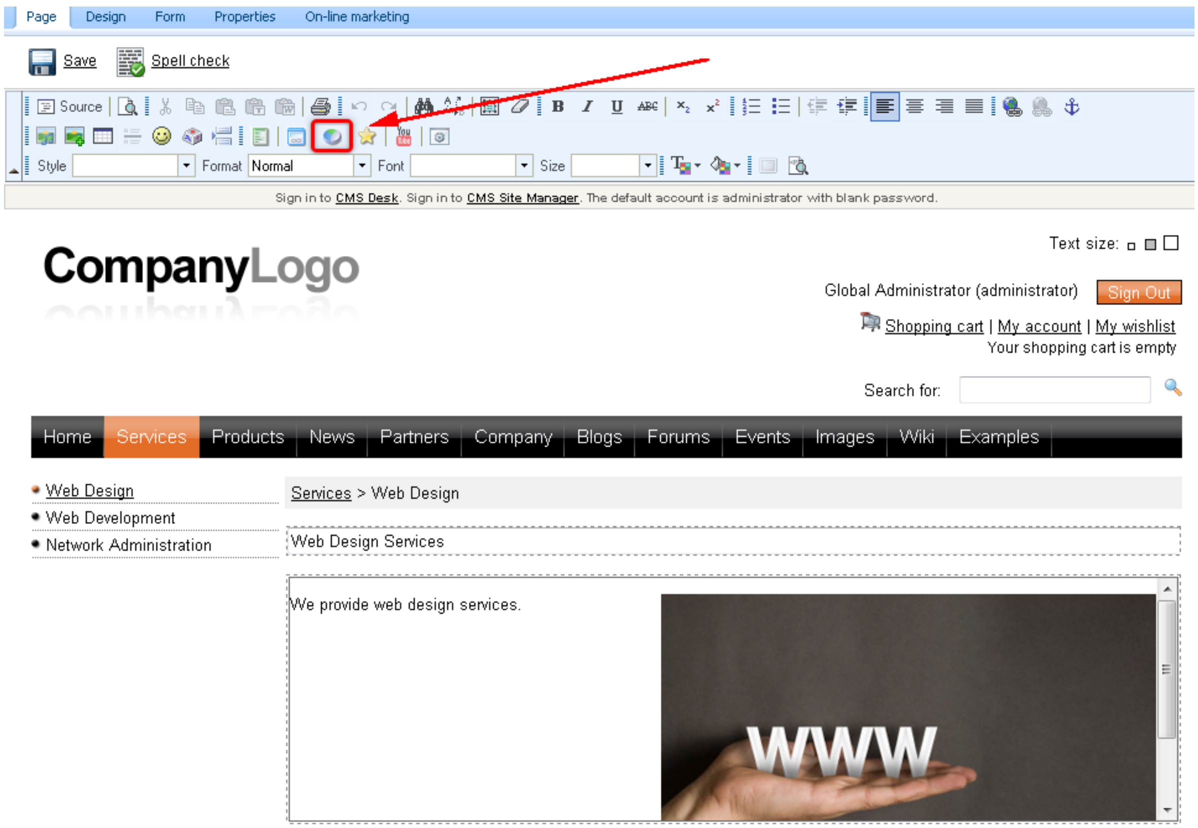
- **Content editors** can place the poll on the page using the **Insert Poll** button in the WYSIWYG editor toolbar
- **Developers** can place the poll on the page using the **Polls\Poll** web part; this options provides more configuration options

Publishing polls for content editors

After defining the poll, you need to go to the **CMS Desk -> Content** section and edit the page where you want to display the poll. Edit the page content using the built-in WYSIWYG editor and click the **Insert Poll** button in the editor toolbar. Then you can choose the poll you want to display from the list of polls.

The editor inserts special expression like this at the cursor position:

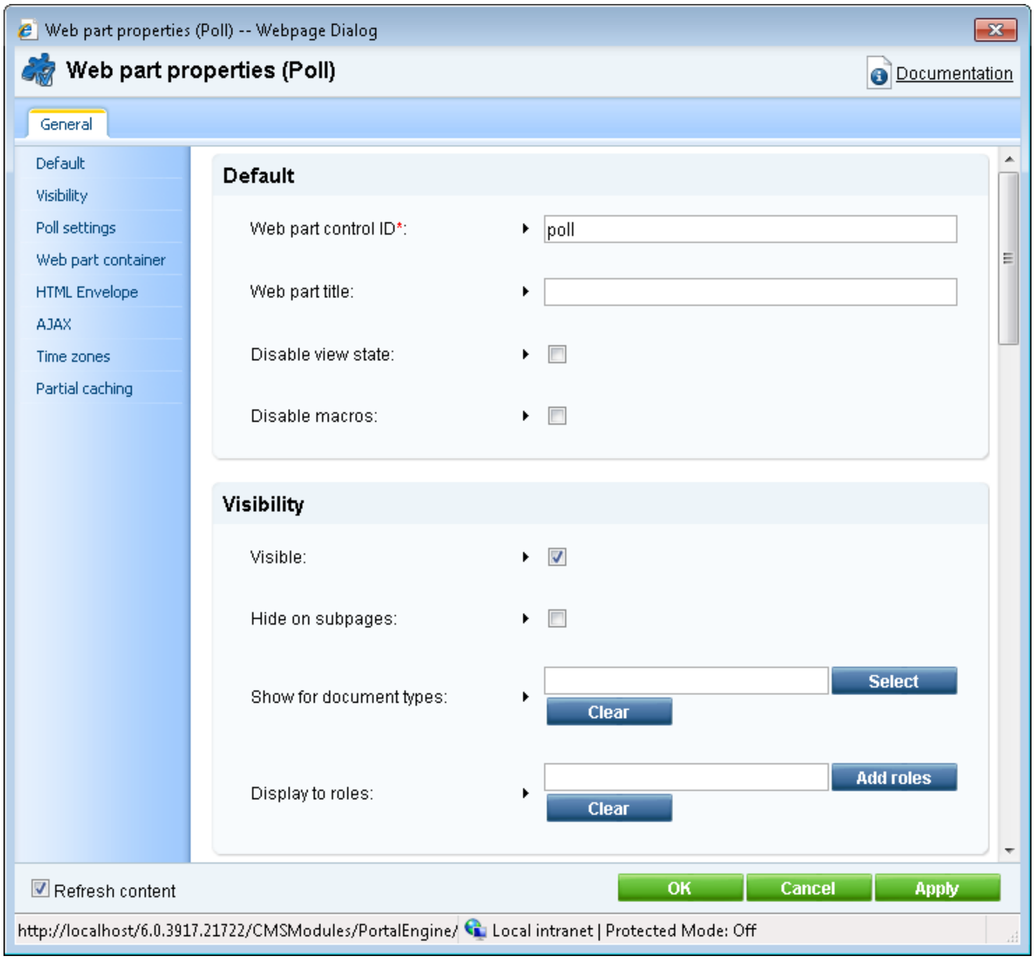
- *%%control:PollControl?ProductSurvey%%*



Publishing polls for developers

If you are a developer, you can go to **CMS Desk -> Content ->** ... choose the page where you want to put the poll ... -> click **Design** and add the **Polls\Poll** web part on the page.

You need to enter the name of the poll. Then you can configure some additional settings of the poll that are described in the Kentico CMS Web Parts reference.



If you are using **ASPX page templates**, you need to drag-and-drop the user control (web part) *CMSWebParts\Polls\Poll.ascx* on your ASPX page.

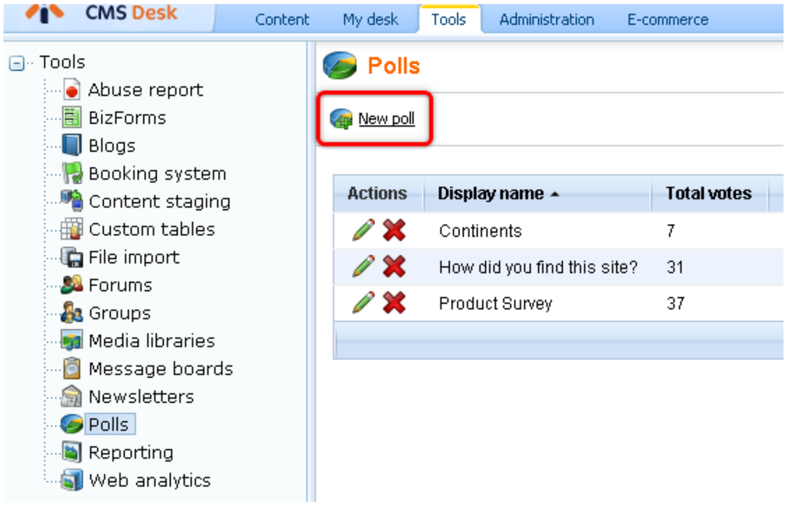
Tip: Poll not showing up in the list

If the poll is not showing up in the list of polls and you cannot insert it, you may need to enable the poll for the given website. You can do that in **CMS Desk -> Tools -> Polls**. Click the **Sites** tab and assign the poll to the required site.

8.31.4 Adding a poll to your site

Here you will learn how to create a new poll and publish it on a page on your website. Please note that you will have to have the Corporate Site sample website installed in order to follow these steps.

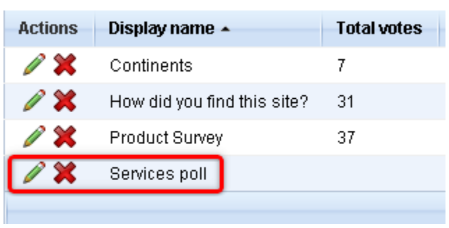
1. First of all you will need to create a new poll. Go to **CMS Desk -> Tools -> Polls** and click the  **New poll** link.



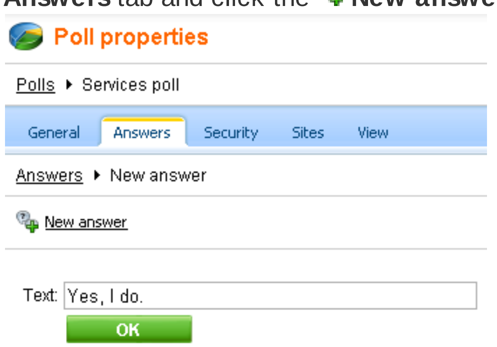
2. Enter the following information for the new poll.

- **Display name:** Services poll
- **Code name:** ServicesPoll
- **Title:** Services poll
- **Question:** Do you like our services?

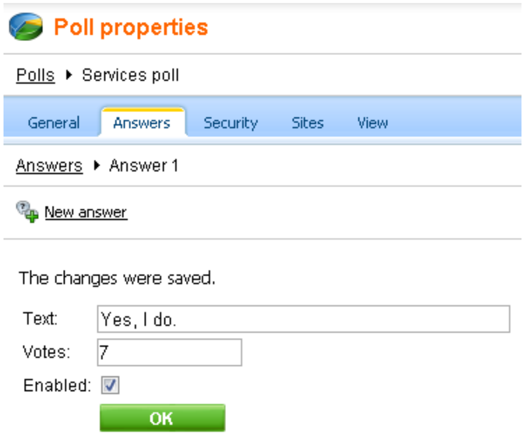
Click **OK** to save the changes. If you switch back to the **CMS Desk -> Tools -> Polls** tab, the poll you have just created will be displayed in the list.



3. Now you need to define some answers for your new poll. In the **Poll properties** dialog, switch to the **Answers** tab and click the **New answer** link. Enter *Yes, I do.* into the **Text** text box and click **OK**.

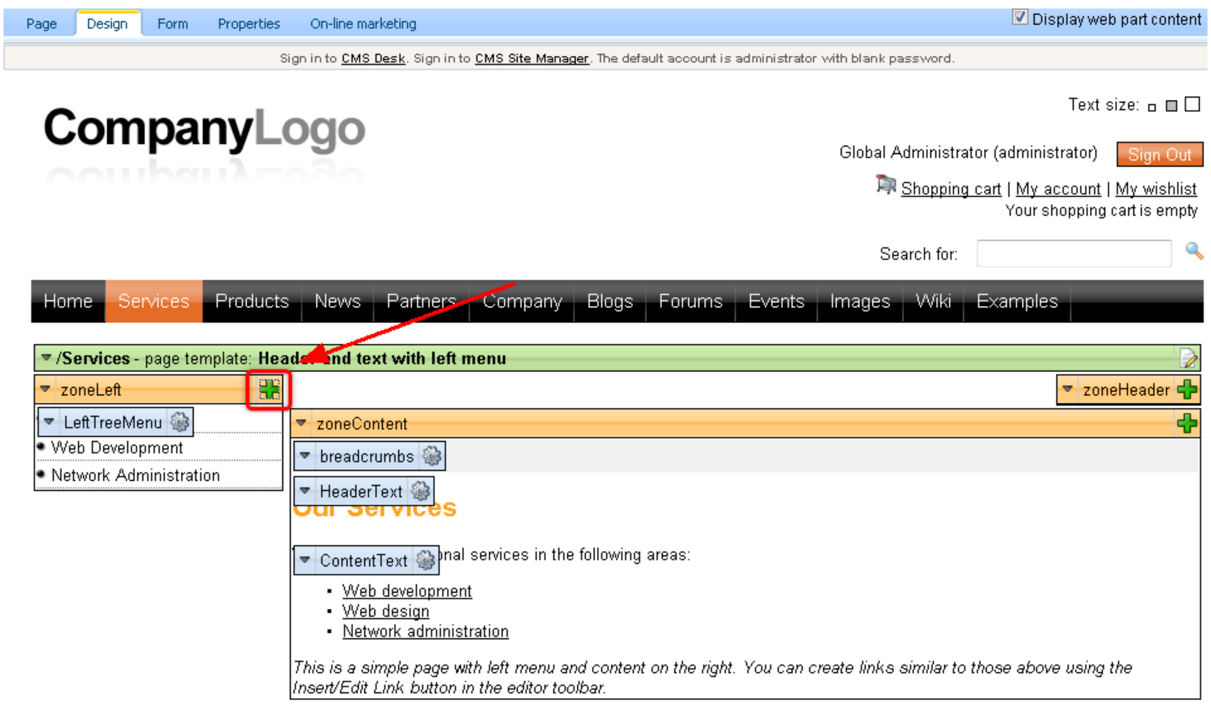


To make the answer available on the live site, check the **Enabled** check box. Besides, you can also enter the initial value for votes. For the purpose of this example, enter 7. Click **OK** to save the changes.

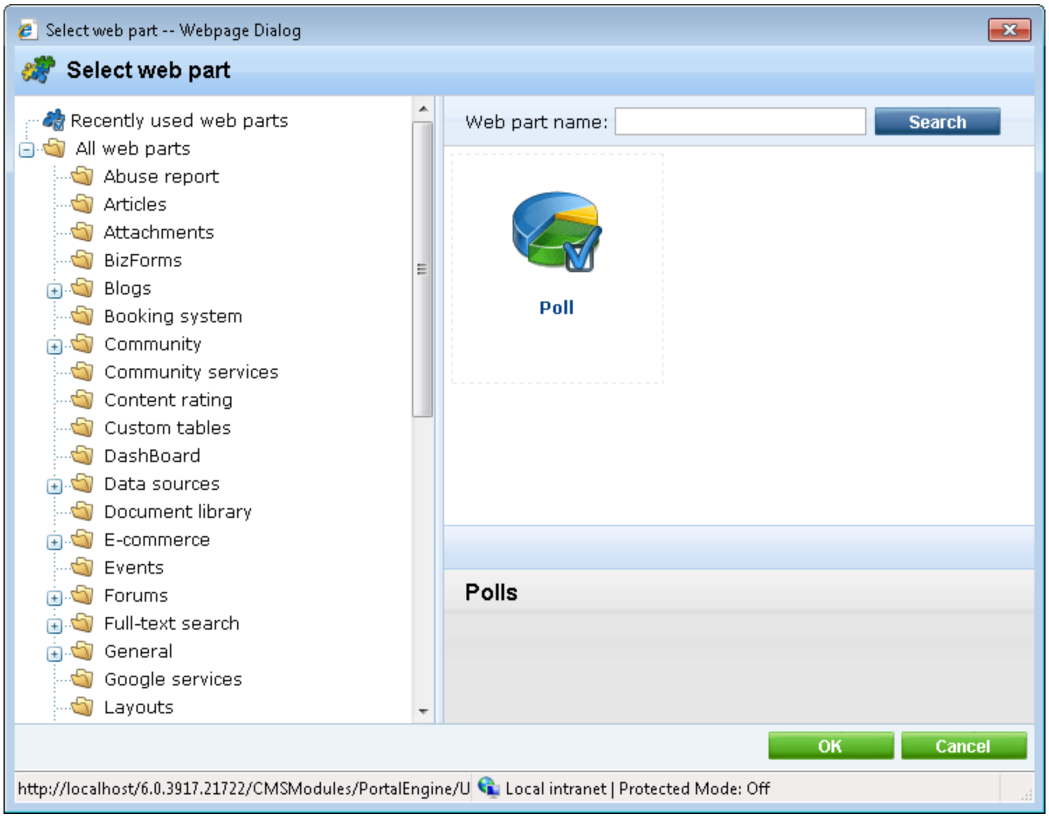


Repeat this step entering *No, I don't.* as an alternative answer and 3 as its initial value for votes. Optionally you can define more answers for your poll.

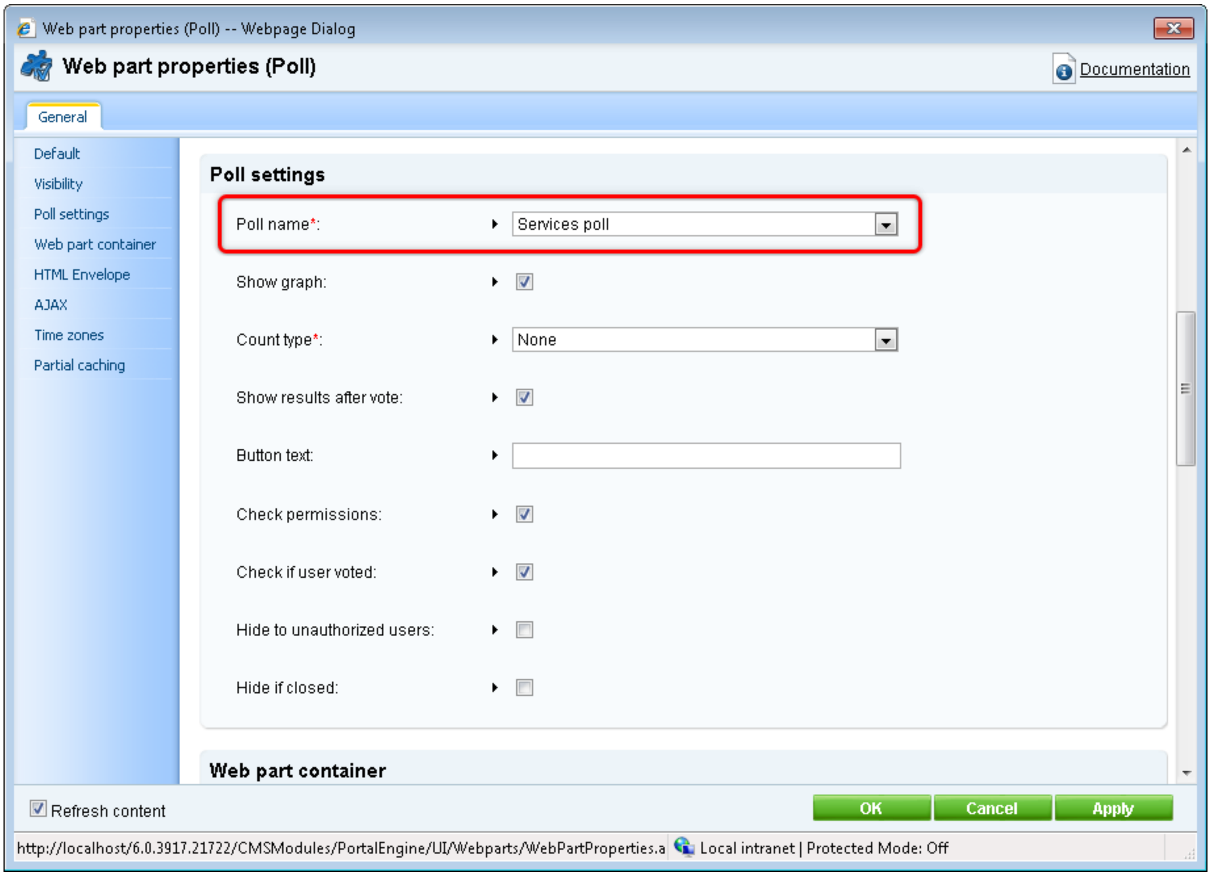
4. Now you are ready to publish your new poll on the website. Go to **CMS Desk -> Content -> Corporate Site -> Services** and switch to the **Design** tab. Click **Add web part** () in the **zoneLeft** web part zone



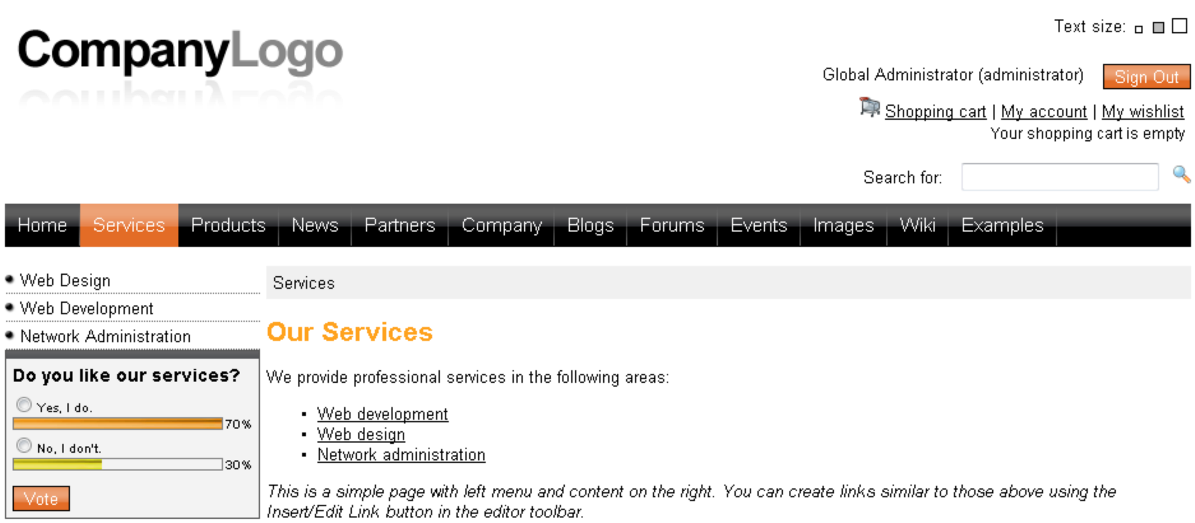
and in the **Select web part** dialog choose the **Polls\Poll** web part.



5. In the **Web part properties** dialog, make sure *Services poll* is selected as **Poll name** and click **OK**. Please note that in this dialog you can also configure some additional settings of the poll that are described in the Kentico CMS Web Parts reference.



Congratulations, you have just learned how to add a new poll to your website.



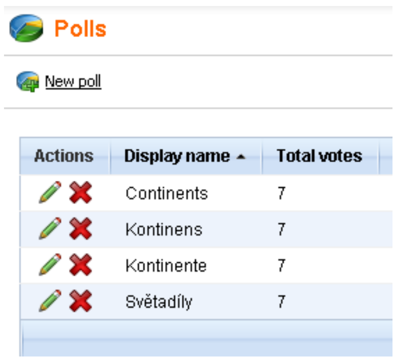
Please note

> You can also add a poll to you web page using the WYSIWYG editor as described in the Publishing polls topic. However, this option gives you fewer possibilities of further adjustment of your poll.

8.31.5 Multilingual support

When you use the Polls module on a multilingual website, you can choose from two options:

- You can create **a new poll for every language** - this is useful if you wish to track votes for different cultures/countries of your website.



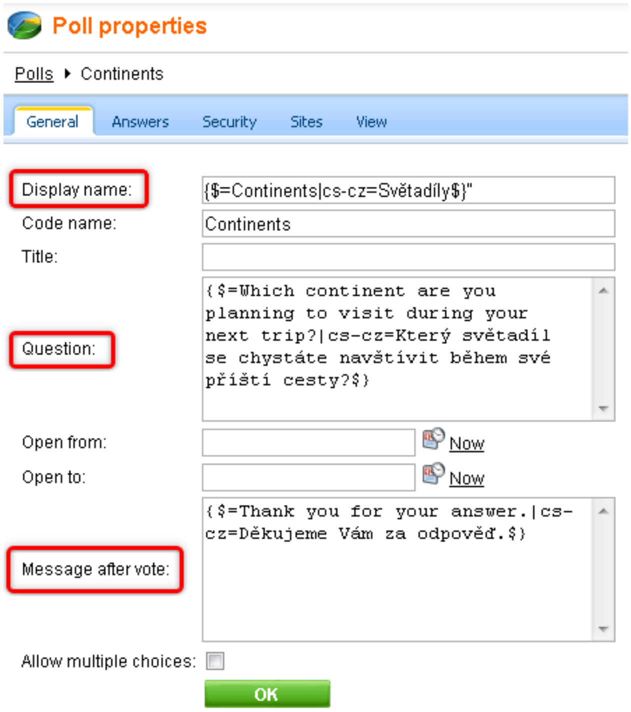
or

- You can create **a single poll for all languages** using in-place localization expressions - in this case, all votes are tracked together. In order to learn how to localize strings of the poll, please refer to the Localization Expressions topic. More detailed information on these expressions can be found in the Appendix A - Macro expressions topic.

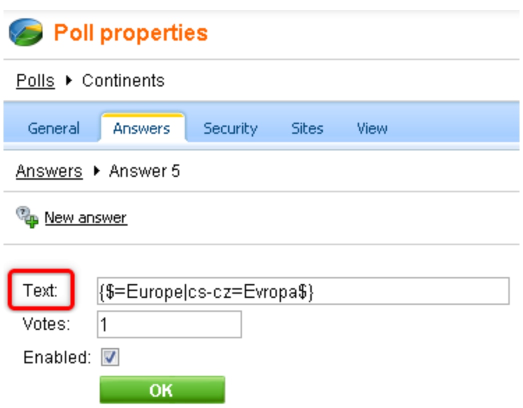
If you want to create a single localized poll (or adjust an existing one to create a single localized poll), you will have to take steps similar to those described in the Managing polls topic. The main difference consists in entering localization expressions into the corresponding text fields. For example, if you would like to have your poll both in the website default language and in Czech, the macro must be entered in the *{$=Default string|cs-cz=Czech string$}* format.

On the **General** tab of the **Poll properties** dialog, use localization expressions for the following fields:

- Display name
- Question
- Message after vote



Click **OK** to save the changes. Now switch to the **Answers** tab to localize the answers of your poll. You will use the localization expression only in the **Text** text field.



When you have entered the new value, click **OK** again.

Please note

To culturally localize your polls, you can also use basic localization macros. These are entered in the *{$string.key$}* format. The system uses the *ResHelper.GetString("string.key")* method and replaces the macros with the appropriate resource strings. To learn more about this type of localization, please refer to the Configuring multilingual and RTL UI topic.

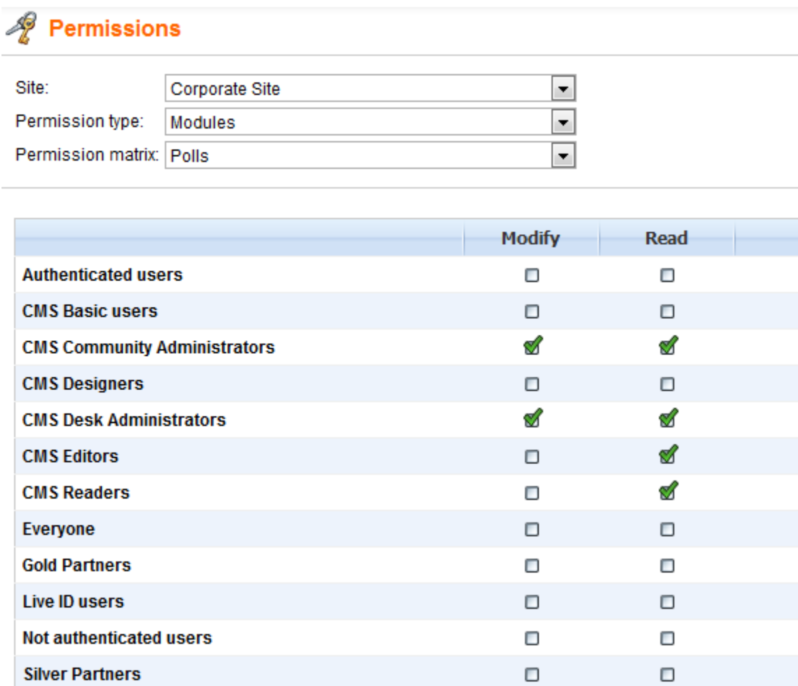
8.31.6 Security

Access to the Polls module

The access to the Polls module can be configured in **CMS Desk -> Administration -> Permissions ->** ... select the **Modules -> Polls** permission matrix.

The following permissions can be assigned to particular roles:

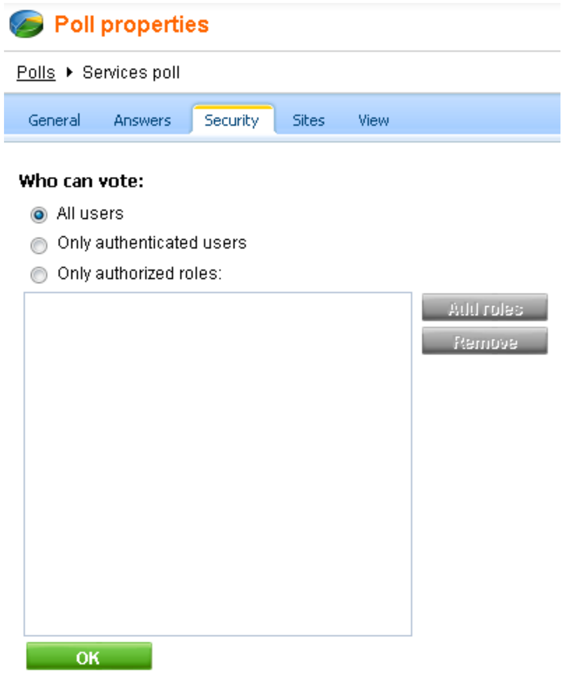
- **Modify** - members of the role can create, edit and delete polls
- **Read** - members of the role can view polls and their configuration, but are not allowed to make any changes to them



Access to voting

You can configure the poll and specify which users are authorized to vote on the **Security** tab of the **Poll properties** dialog. Here you can choose one of the following options:

- **All users** - any visitor can vote
- **Only authenticated users** - only site members who sign in can vote
- **Only authorized roles** - only authenticated members of chosen roles can vote



The developers can also customize the behavior of the **Polls** web part using the following web part properties:

- **Check permissions** - indicates if permissions for voting specified for the given poll should be checked
- **Hide to unauthorized users** - hides the web part if the user is not authorized to vote

8.31.7 Polls internals and API

8.31.7.1 Database tables and API classes

The Polls module uses the following database tables:

- **Polls_Poll** - polls
- **Polls_PollAnswer** - answers for polls (1:N)
- **Polls_PollRoles** - roles authorized to vote (M:N)
- **Polls_PollSite** - sites where the poll can be used (M:N)

The polls API is provided by the **CMS.Polls** namespace classes:

- **PollAnswerInfo, PollAnswerInfoProvider** - these classes provide functionality for managing answers
- **PollInfo, PollInfoProvider** - these classes provide functionality for managing polls

The following topics show examples of how these classes can be used:

- Creating a new poll
- Updating an existing poll
- Add a new answer
- Change the number of votes

- Add authorised roles
- Deleting a poll
- Deleting a poll answer

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.31.7.2 Creating a new poll

The following sample code shows how you can create a new poll using the API:

[C#]

```
using CMS.Polls;

...

        // Create new poll info
        PollInfo pollObj = new PollInfo();

        // Set the fields
        pollObj.PollCodeName = "testingSitePoll";
        pollObj.PollDisplayName = "Testing site poll";
        pollObj.PollTitle = "Your free time";
        pollObj.PollQuestion = "What do you do in your free time?";
        pollObj.PollResponseMessage = "Thank you for your vote.";
        pollObj.PollAllowMultipleAnswers = false;
        pollObj.PollAccess = SecurityAccessEnum.AllUsers;
        pollObj.PollOpenFrom = DateTime.Now;
        pollObj.PollOpenTo = DateTime.Now.AddMonths(1);

        // Save the object
        PollInfoProvider.SetPollInfo(pollObj);

        // Add poll to current site
        PollInfoProvider.AddPollToSite(pollObj.PollID, CMSContext.CurrentSiteID);
```

8.31.7.3 Updating an existing poll

The following sample code shows how you can modify poll properties using the API:

[C#]

```
using CMS.Polls;

...
```

```
        string pollName = "testingSitePoll";

        // Get poll info
        PollInfo pollObj = PollInfoProvider.GetPollInfo(pollName);

        if (pollObj != null)
        {
            // Set the field values
            pollObj.PollResponseMessage = "Thank you for your time.";
            pollObj.PollAllowMultipleAnswers = true;
        }

        // Save the object
        PollInfoProvider.SetPollInfo(pollObj);
```

8.31.7.4 Add a new answer

The following sample code shows how you can create a new answer for an existing poll using the API:

[C#]

```
using CMS.Polls;

...

        string pollName = "testingSitePoll"; // Poll code name

        // Get poll info for the specified code name
        PollInfo pollObj = PollInfoProvider.GetPollInfo(pollName);

        if (pollObj != null)
        {
            // Create a new answer object
            PollAnswerInfo pollAnswerObj = new PollAnswerInfo();

            // Set the fields
            pollAnswerObj.AnswerOrder = PollAnswerInfoProvider.GetLastAnswerOrder(po
            pollAnswerObj.AnswerCount = 0;
            pollAnswerObj.AnswerPollID = pollObj.PollID;
            pollAnswerObj.AnswerEnabled = true;
            pollAnswerObj.AnswerText = "Yes, I love it.";

            // Save the answer
            PollAnswerInfoProvider.SetPollAnswerInfo(pollAnswerObj);
        }
```

8.31.7.5 Change the number of votes

The following sample code shows how you can increase the number of votes for a given answer using the API. It's useful if you want to create your own voting dialog.

[C#]

```
using CMS.Polls;

...

       string pollName = "testingSitePoll"; // Poll code name

        // Get poll info for the specified code name
        PollInfo pollObj = PollInfoProvider.GetPollInfo(pollName);

        if (pollObj != null)
        {
            // Get all answers of the poll
            DataSet ds = PollAnswerInfoProvider.GetAnswers(pollObj.PollID);

            if (!DataHelper.DataSourceIsEmpty(ds))
            {
                // Get first pollanswerinfo object
                PollAnswerInfo pollAnswerObj = new PollAnswerInfo(ds.Tables[0].Rows[

                if (pollAnswerObj != null)
                {
                    // Increment votes by 1
                    pollAnswerObj.AnswerCount++;

                    // Save the data
                    PollAnswerInfoProvider.SetPollAnswerInfo(pollAnswerObj);
                }
             }
          }
```

8.31.7.6 Add authorized roles

The following sample code shows how you can add roles authorized for voting.

[C#]

```
using CMS.Polls;

...

        string pollName = "testingpoll"; // Poll code name
        string roleName = "cmseditor"; // Role code name
        string siteName = "CMSTestingSite"; // Role's site code name

        // Get poll info for the specified code name
        PollInfo pollObj = PollInfoProvider.GetPollInfo(pollName);

        // Get role info for specified role name and site name
```

```
        RoleInfo roleObj = RoleInfoProvider.GetRoleInfo(roleName, siteName);

        if ((pollObj != null) && (roleObj != null))
        {
            // Change poll access to access for specified roles only
            pollObj.PollAccess = PollInfoProvider.ACCESS_ROLES;

            // Save the object
            PollInfoProvider.SetPollInfo(pollObj);

            // Check if the role has already been added
            if (!PollInfoProvider.IsRoleAllowedForPoll(pollObj.PollID, roleName))
            {
                // Add role to the poll
                PollInfoProvider.AddRoleToPoll(roleObj.RoleID, pollObj.PollID);
            }
         }
```

8.31.7.7 Deleting a poll

The following sample code shows how you can delete a poll including its answers.

[C#]

```
using CMS.Polls;

...

        string pollName = "testingSitePoll";

        // Get poll info for the specified code name
        PollInfo pollObj = PollInfoProvider.GetPollInfo(pollName);

        // Delete the poll
        PollInfoProvider.DeletePollInfo(pollObj);
```

8.31.7.8 Deleting a poll answer

The following sample code shows how you can delete a poll answer.

[C#]

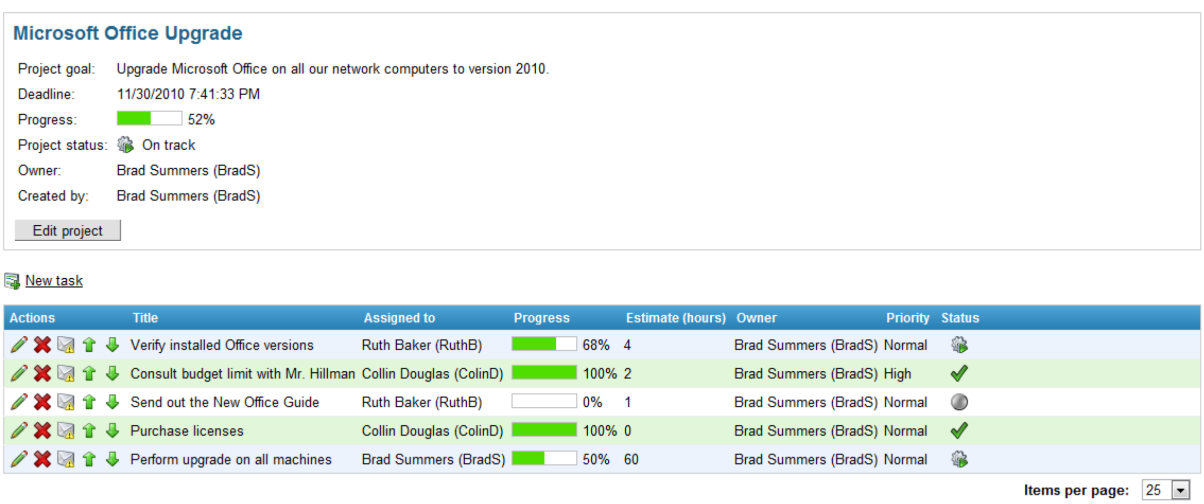
```
using CMS.Polls;

...

        string pollName = "testingSitePoll";

        // Get poll info for the specified code name
```

```
            PollInfo pollObj = PollInfoProvider.GetPollInfo(pollName);

            if (pollObj != null)
            {
                // Get all poll answers
                DataSet answers = PollAnswerInfoProvider.GetAnswers(pollObj.PollID);

                if (!DataHelper.DataSourceIsEmpty(answers))
                {
                    // Create pollanswerinfo object from datarow
                    PollAnswerInfo pollAnswerObj = new PollAnswerInfo(answers.Tables[0].

                    if (pollAnswerObj != null)
                    {
                        // Delete poll answer
                        PollAnswerInfoProvider.DeletePollAnswerInfo(pollAnswerObj);
                    }
                 }
              }
```

8.32 Project management

8.32.1 Overview

Project management is a module that provides a system that helps organize work and keep track of its progress. This is achieved by creating objects within the Kentico CMS system called **projects**, that represent a certain segment of work with a specific goal, time frame, set of conditions or various other properties. Individual assignments called **tasks**, which are usually (but not necessarily) categorized under a certain project, can be given to users within the system.



Projects can be created, monitored and managed on any page (document) in the site content tree that contains the appropriate project management web parts. The document where a project is created is important, as the project is bound to that page and can only be viewed or edited on that specific page. This functionality can be used to categorize projects by having several different pages with project management web parts, which can then be individually configured.

All projects and their tasks can be administered at **CMS Desk -> Tools -> Project Management**, regardless of the document where they were created. Here, administrators can also implement custom priority and status options to be used by projects and tasks. More detailed information about administering and using projects and tasks is given in the Managing projects and tasks topic.

Project management is also integrated into the groups module. Group projects function just as described above, but they belong to a certain group in addition to being bound to the specific group page where they were created. Group projects only operate within the scope of their group, which means that tasks can only be assigned to group members, project security settings can be configured for group roles instead of site roles etc. This can be used to create groups for handling certain types of projects. Projects associated with a group are not displayed in the **CMS Desk -> Tools -> Project Management** interface. They can instead be managed by group or global administrators under the **Projects** tab when editing a group.

Users can manage tasks that they own or those that are assigned to them on pages containing specific project management web parts or widgets. Tasks are displayed without regard to the project they belong under or the page or group where they were created, the only thing that matters is that they are relevant to the current user (and that any permission requirements are fulfilled). If the displaying web part/widget is configured to do so, even tasks from other sites can be viewed. These web parts and widgets also allow users to create new personal tasks that aren't categorized under any group or project.

To ensure that users are aware of task assignments or any changes in tasks related to them, project management uses a system of notification e-mails based on templates, that are automatically sent when required.

An overview of the web parts and widgets that allow users to interact with the project management module on the live site can be found in the Project management web parts and widgets topic.

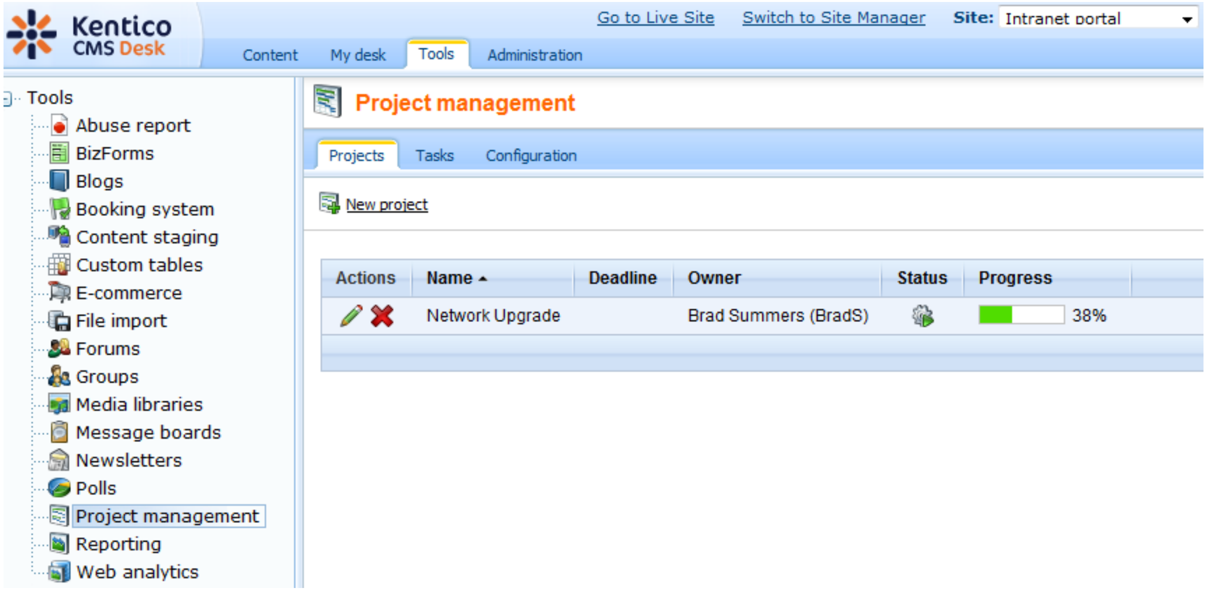
Permissions for project/task management are determined by many factors. Please refer to the Security topic for more information.

8.32.2 Managing projects and tasks

There are several ways how projects and their tasks can be used and managed:

Using the main project management interface

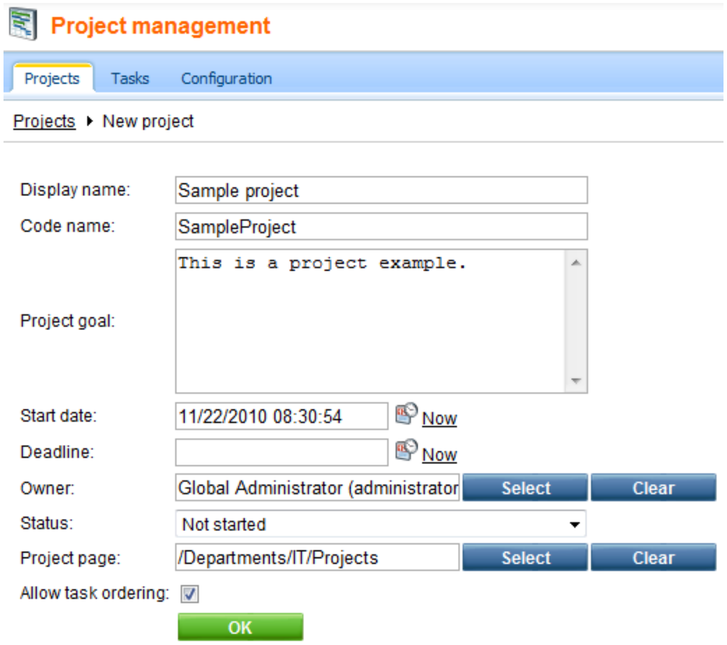
All projects that do not belong to a group can be managed by users with sufficient permissions at **CMS Desk -> Tools -> Project management -> Projects**.



Here, existing projects are displayed in a list where they may be **Edited** () or **Deleted** ().

Create a new project using the **New project** link and fill in the following properties:

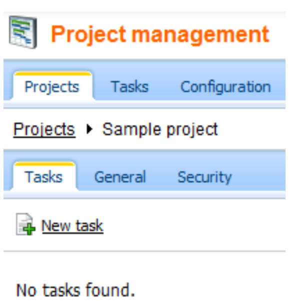
- **Display name** - Name of the project that is displayed on the live site and in the user interface.
- **Code name** - Code name of the project.
- **Project goal** - Text description of the purpose of the project.
- **Start date** - Specifies the date and time when work on the project should start.
- **Deadline** - Specifies the date and time before which the project should be complete.
- **Owner** - Allows the selection of the user who will be set as the project owner. The owner is usually the user who ensures that the project is completed successfully. By default, the user who created the project is entered.
- **Status** - Allows one of the predefined project statuses to be selected. Statuses are used to indicate which life cycle step the project is currently in and can be managed on the *Configuration* tab of this interface.
- **Project page** - Allows the selection of a document from the current site to which the project will be bound.
- **Allow task ordering** - If enabled, the order of tasks under the project can be changed using arrow buttons.



Click the **OK** button.

This will open the editing interface of the new project, where three tabs are available. The **General** tab allows the properties that were set when the project was created to be modified and the **Security** tab is where permissions for the project can be configured (learn more in the Security topic).

The **Tasks** tab contains a list of all tasks under the project, which will currently be empty.

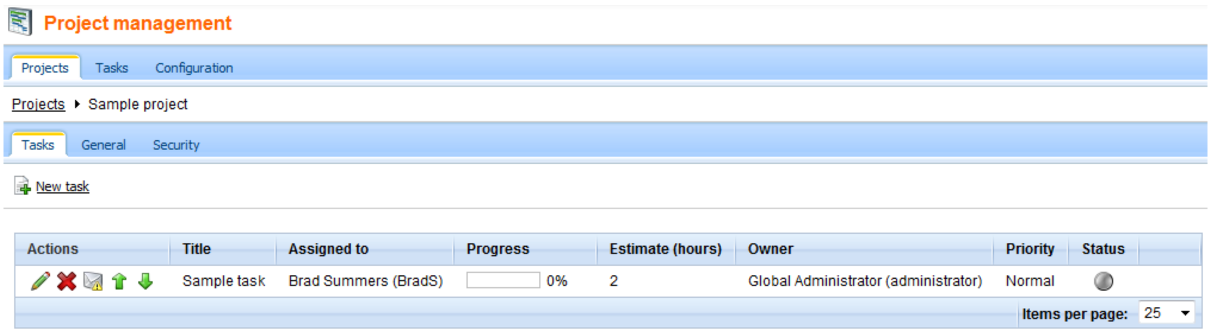


Click the **New task** link and enter the following properties:

- **Title** - Name of the task that is displayed on the live site and in the user interface.
- **Owner** - Allows the selection of the user who will be set as the task owner. The owner is usually the user who checks that the task was completed correctly. By default, the user who created the task is entered.
- **Progress** - A percentage representing the amount of progress that has been made on the task.
- **Estimate** - Is used to set an approximate amount of hours that the completion of the task should take.
- **Deadline** - Specifies the date and time before which the task should be complete.
- **Status** - Allows one of the predefined task statuses to be selected. Statuses are used to indicate which life cycle step the task is currently in.
- **Priority** - Allows one of the predefined task priorities to be selected.

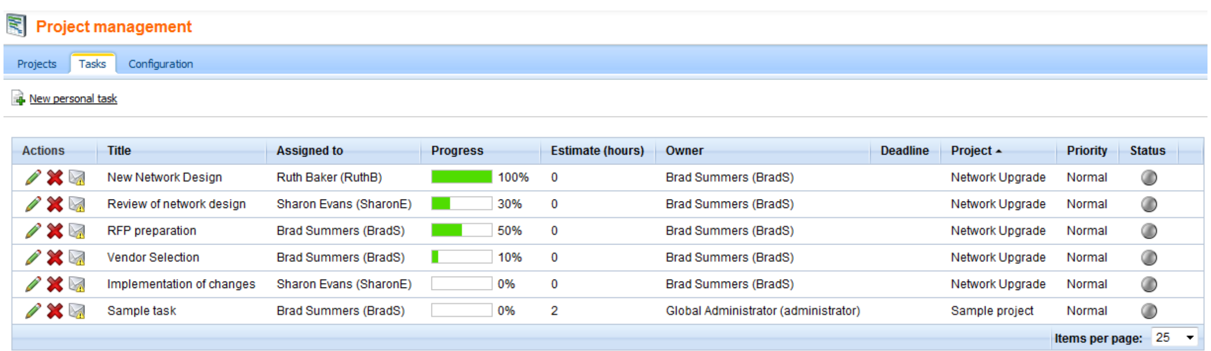
- **Is private** - If enabled, the task will only be displayed to its owner, the user to whom it is assigned and to users who have permissions to manage the project that the task belongs under (if applicable).
- **Assigned to** - Allows the selection of the user who should complete this task. A notification e-mail is automatically sent to the specified user when the task is created or modified.
- **Description** - Text containing the goal of the task and any other relevant information.

Click **OK**.

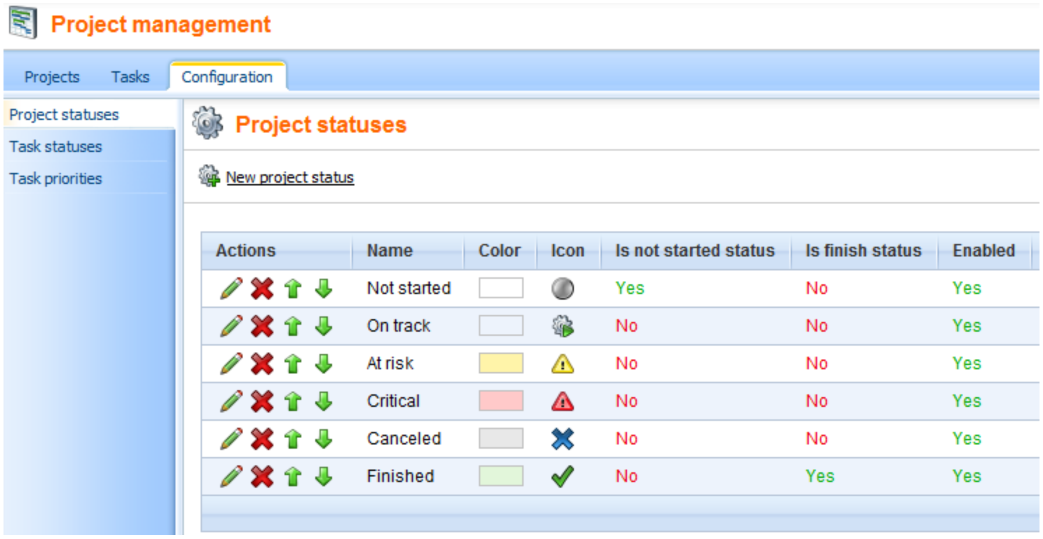


You can manage existing tasks using the appropriate action buttons on the left side: **Edit** () or **Delete** (). The **Send reminder** () button allows an e-mail to be sent to the user to whom the task is assigned, reminding them about the task. If the **Allow task ordering** property of the edited project is enabled, the **Move up** () and **Move down** () buttons can be used to change the order of the tasks.

The **Tasks** tab on the top level of the project management interface is very similar to to the list of tasks shown when editing a project, except that all tasks are displayed regardless of the project they belong under, including personal tasks without any project.



The **Configuration** tab is where the statuses and priorities that can be set for projects or tasks may be defined. It may only be accessed by users belonging to roles with the **Manage configuration** permission for the project management module (more information can be found in the Security topic). All the usual actions such as **Edit** () and **Delete** () are available.

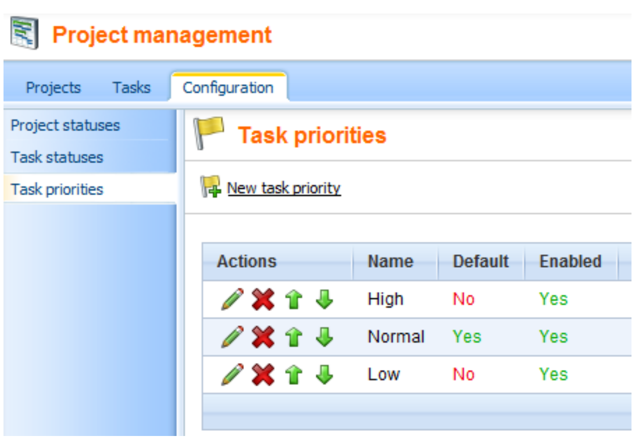


When creating or editing a **status**, the following properties can be set:

- **Display name** - Name of the status that is used in the status selection list.
- **Code name** - Code name of the status.
- **Status color** - Determines the color of projects/tasks that have this status selected. The color selector can be used here.
- **Status icon** - Contains the path to an image used as the icon of the status.
- **Is not started status** - If enabled, projects/tasks with this status are marked as not yet started.
- **Is finish status** - If enabled, projects/tasks with this status are marked as finished. A status cannot be both *not started* and *finished*.
- **Enabled** - If enabled, the status can be selected for projects.

The *not started* or *finished* flag given to a project or task by its status can have an effect on whether it is displayed by certain project management web parts or widgets, depending on their configuration.

The **Task priorities** tab is where priorities, which can be selected to determine how urgent a task is, may be managed.



The following properties are available for task priorities:

- **Display name** - Name of the priority that is used in the task priority selection list.

- **Code name** - Code name of the task priority.
- **Default** - If enabled, the priority is selected by default when a new task is created. Only one priority can be set as default.
- **Enabled** - If enabled, the priority can be selected for tasks.

E-mail notifications

As mentioned above, there are several types of notification e-mails that are sent to users informing them about events related to tasks that they are involved in. These e-mails are sent from the address specified in the value of the **Site Manager -> Settings -> Project management -> Send project management e-mails from** setting and their format can be customized at **Site Manager -> Development -> E-mail templates**. The following e-mail templates are available:

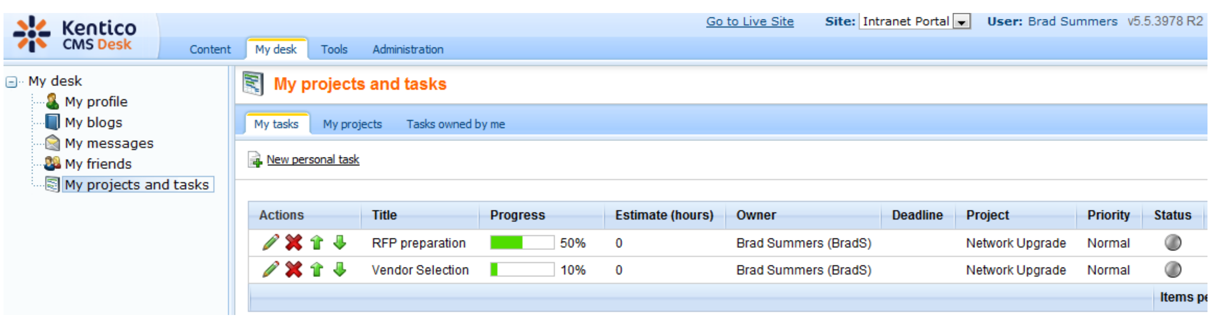
- **Project Management - New task** - Sent when a new task is assigned to the user.
- **Project Management - Changed task** - Sent when an assigned or owned task is in some way modified.
- **Project Management - Overdue task** - Sent when an assigned task reaches its deadline before it is finished.
- **Project Management - Task reminder** - Sent when the **Send reminder** ( ) button is used for an assigned task.

Using the Group management interface

Group projects can be administered by users, who have permissions to manage a given group, by editing the group on its **Projects** tab either at **CMS Desk -> Tools -> Groups** or on-site using the **Group profile** web part. Exactly the same options are available as when using the **Projects** tab of the main project management interface described above, except that the context of the current group is used e.g. for selecting users to whom a task should be assigned.

Using CMS Desk -> My Desk

Users with access to **CMS Desk** can use the **My Desk -> My projects and tasks** interface to manage projects and tasks that they are involved in.



The current user can see a list of all tasks assigned to them on the **My tasks** tab, all the projects that they have permissions to access on the **My projects** tab and all tasks that they own on the **Tasks owned by me** tab. Projects and tasks can be managed using the actions described above.

8.32.3 Project management web parts and widgets

The web parts described in this topic can be placed on pages to create an interface that allows users to work with projects and tasks directly on the live site. They can all be found in the **Project management** web part category. Additional information about using these web parts on the live site is given in the Using project management on the live site topic.

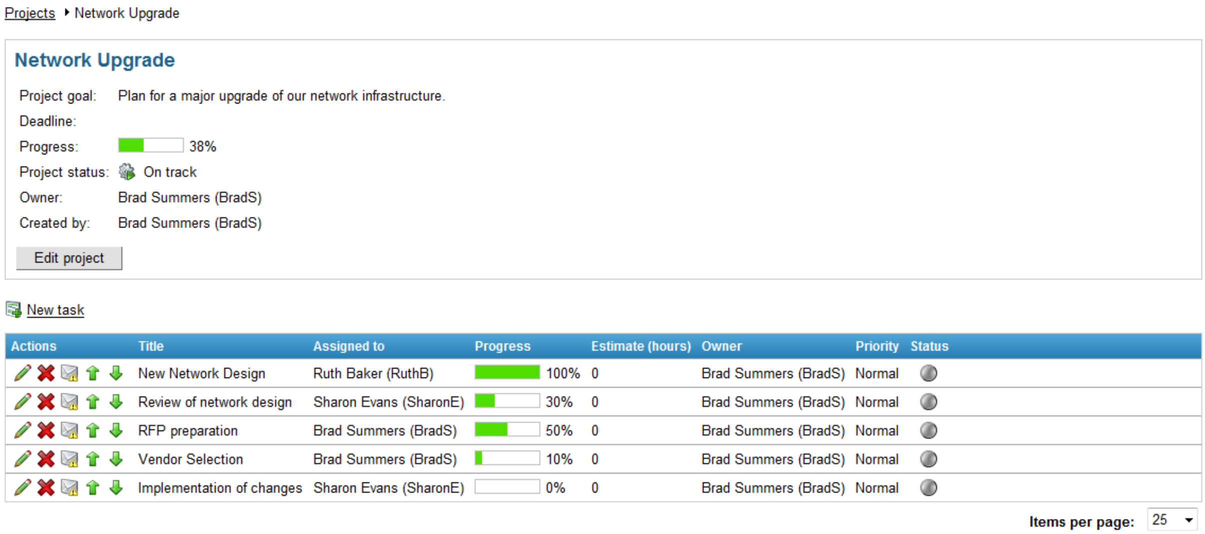
Project list

The Project list web part displays all projects that are bound to the document it is placed on and allows new ones to be created. If the document is owned by a group, only projects that belong under that group are shown.

Specific properties:

- **Show finished projects** - If enabled, projects that have a status designating a finished project are also displayed.
- **Project can be managed by** - Can be used to set the *Create*, *Modify* and *Delete* permissions for the displayed projects. The selection made here overrides the security settings of individual projects. The following options can be selected:
  - *Module administrators* - each displayed project uses its individual security settings
  - *All users* - all site users receive permissions for the displayed projects
  - *Authenticated users* - only signed-in users receive permissions for the displayed projects
  - *Group members* - only members of the group that the current document belongs to receive permissions for the displayed projects
  - *Authorized roles* - only members of the roles selected in the *Authorized roles* property receive permissions for the displayed projects
- **Authorized roles** - Can be used to select the roles that should have permissions for the displayed projects if *Authorized roles* is selected in the *Project can be managed by* property.

Users with the appropriate permissions can manage the displayed projects and their tasks directly on the live site using this web part.



My projects

This web part displays a list of all projects that the current user has access permissions for. Only projects that are bound to an existing document are shown, since the projects in the list also serve as links to that document. Documents that have projects bound to them usually also contain the **Project list** web part and if this is the case, the editing interface of the linked project is automatically opened.

Specific properties:

- **Show finished projects** - If enabled, projects that have a status designating a finished project are also displayed.

The **My projects** widget based on this web part is included in the project management module by default.

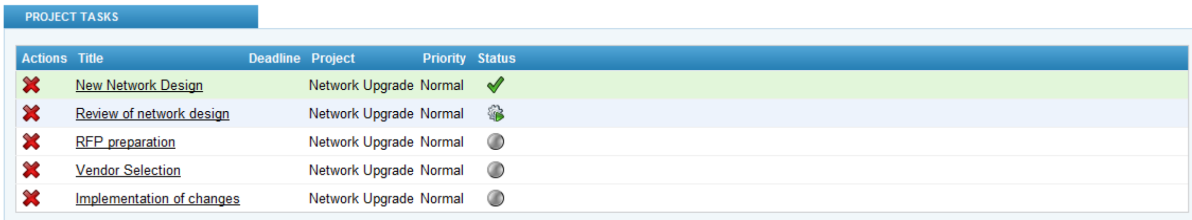
Project tasks

This web part displays a list of tasks from a selected set of projects. Users can only view tasks from those projects that they have access permissions for. Depending on the security settings of the selected projects, tasks can also be edited directly by clicking on their title.

Specific properties:

- **Projects** - Allows the projects whose tasks should be displayed to be selected. All projects from the current site that do not belong under any group are available.
- **Show overdue tasks** - If enabled, tasks that have passed their deadline are displayed.
- **Show on time tasks** - If enabled, tasks that haven't yet reached their deadline are displayed.
- **Show private tasks** - If enabled, private tasks are displayed. They are only visible to those users who are allowed to see them.

- **Show finished tasks** - If enabled, tasks that have a status designating a finished task are displayed.
- **Show status as** - Allows the way that task statuses are displayed to be selected. Possible options are *Icon*, *Text* and *Icon & Text*.
- **Allow task actions** - If enabled, users with the appropriate permissions are allowed to delete the displayed tasks.

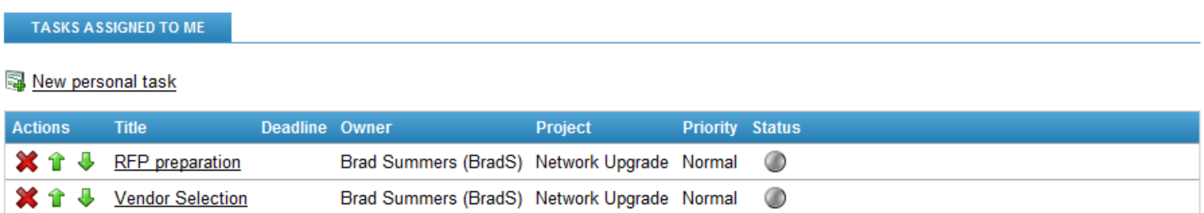


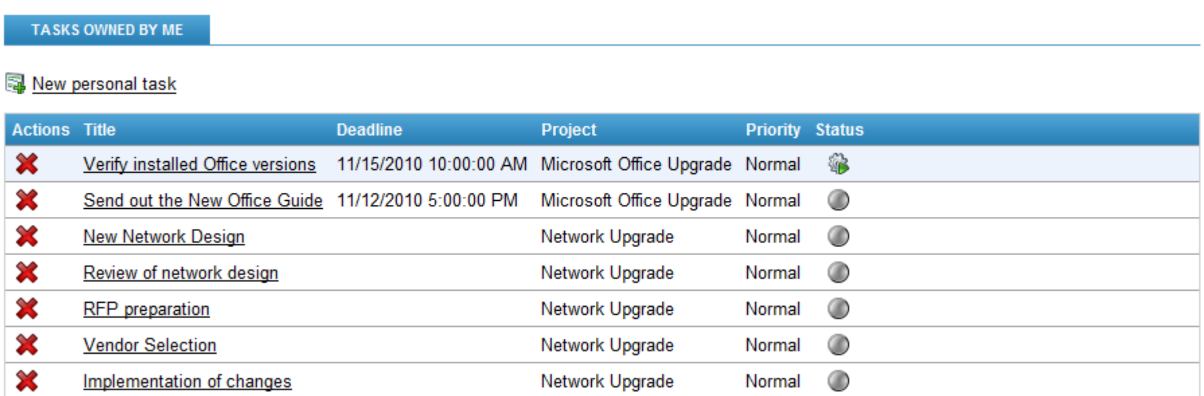
Tasks owned by me / Tasks assigned to me

These two web parts display a list of tasks that are either owned by or assigned to the current user. Personal tasks (those not belonging to a project) are always displayed, but project tasks are only shown if the user has permissions to access the given project. The tasks in the list can be edited directly by clicking their title.

Specific properties:

- **Show overdue tasks**, **Show on time tasks**, **Show private tasks**, **Show finished tasks**, **Show status as** - Are the same as for the *Project tasks* web part described above.
- **Allow task actions** - If enabled, the displayed tasks can be deleted and new personal tasks (not categorized under any project) can be created using the web part.
- **Site** - Can be used to select if tasks should be displayed from *(all sites)*, or only the *(current site)*.





The **Tasks owned by me** and **Tasks assigned to me** widgets based on these two web parts are included in the project management module by default.

Task info panel

The Task info panel can be used to display a message containing task related information, usually showing the amount of tasks that are assigned to the current user serving as a link to a document where the details of the tasks can be viewed.

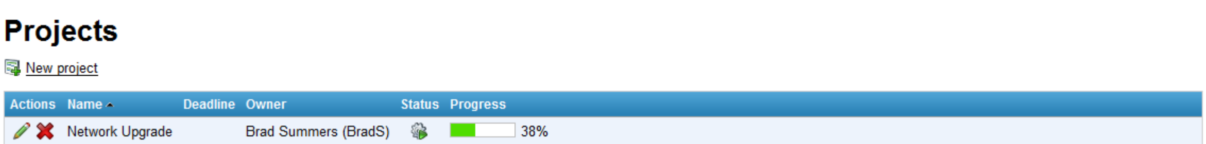
Specific properties:

- **Task detail page URL** - Contains the URL of the document that is linked when the displayed message is clicked. If left empty, the value is taken from the *Site Manager -> Settings -> Project management -> Task detail page* field. In the case that neither of these locations contain a value, the message will not work as a link.
- **Info text** - Contains the text of the displayed message. It may contain the *{0}* formatting macro expression, which will be resolved into the number of tasks assigned to the current user. Sample value: *{0} active task(s)*
- **Include not started tasks** - If enabled, tasks that have a status designating a task that has not been started yet are counted by the web part.
- **Include finished tasks** - If enabled, tasks that have a status designating a finished task are counted by the web part.

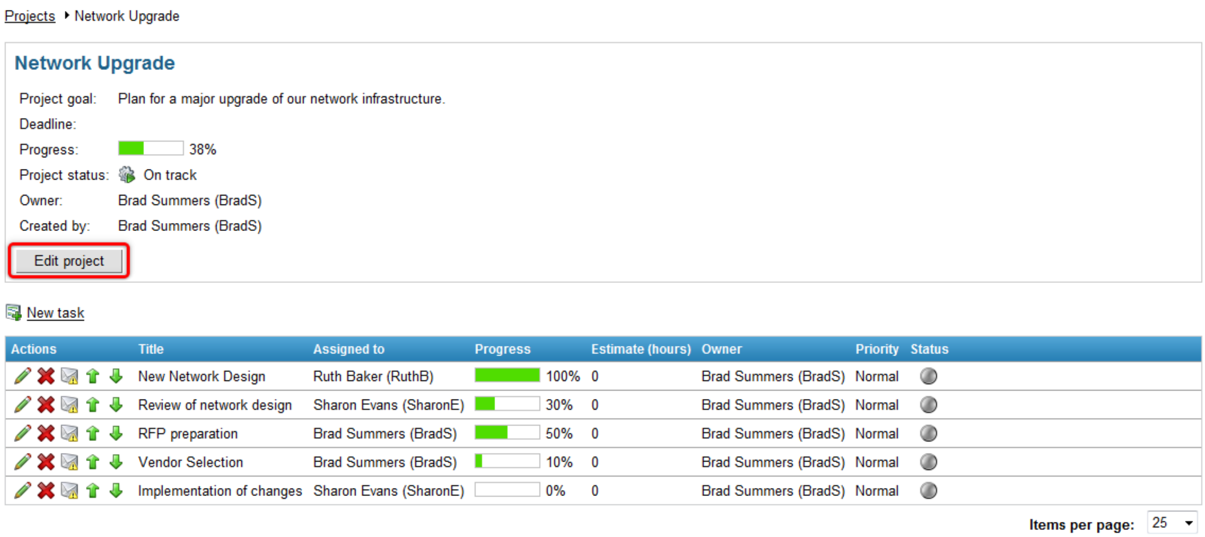
8.32.4 Using project management on the live site

Users can work with projects and tasks on the live site through Project management web parts and widgets. The actions that are available to users depend on project security settings and the configuration of specific web parts or widgets.

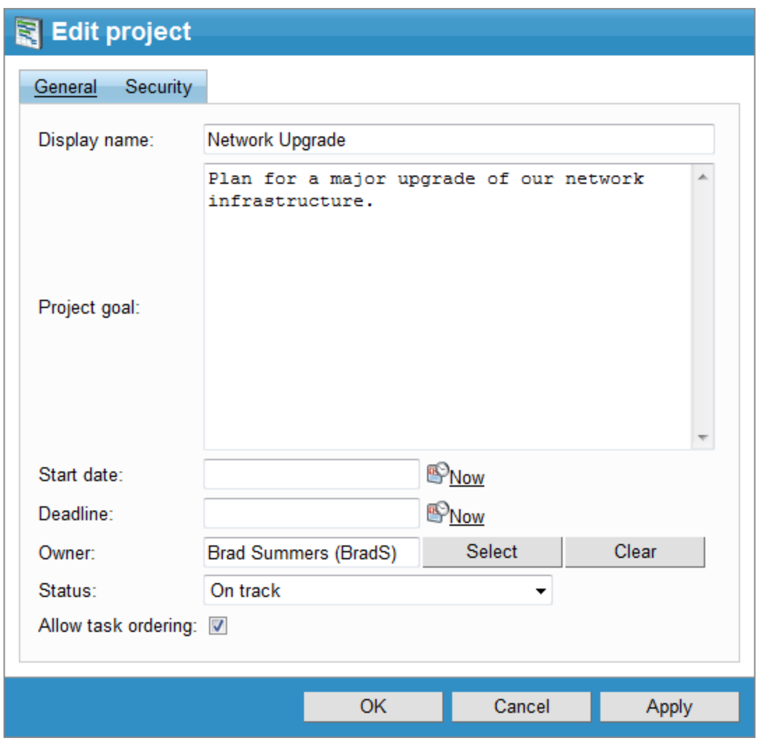
The **Project list** web part allows projects, that are bound to the document it is placed on, to be managed in a similar fashion as in the administration interface, which is described in the Managing projects and tasks topic.



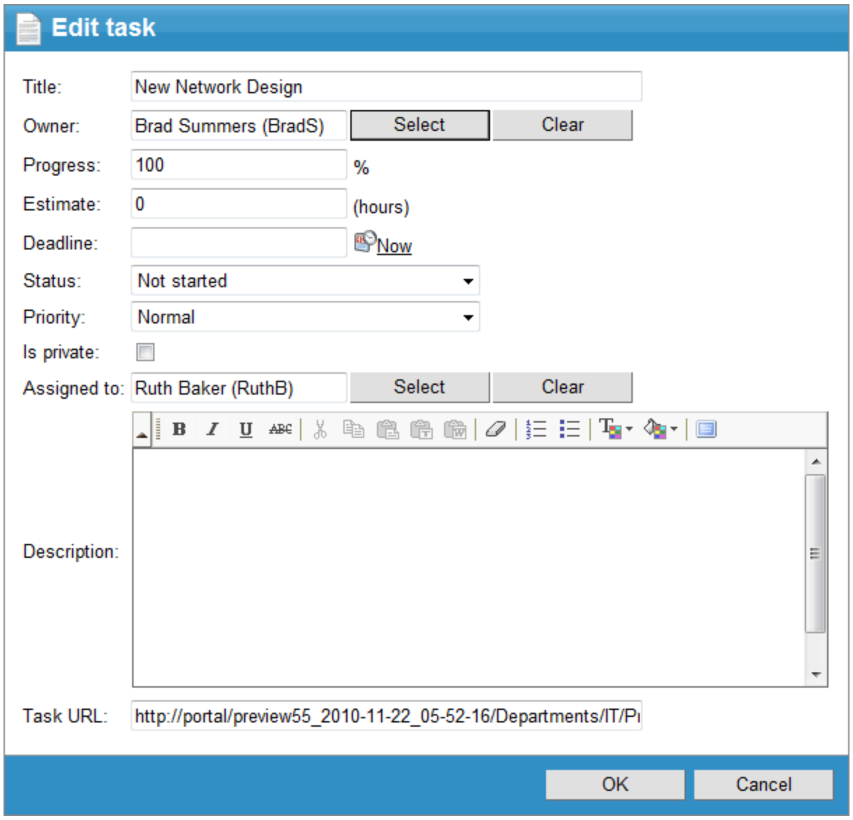
When a project is edited (), its details and the tasks it contains are displayed. Configuration of the project itself can be achieved by clicking the **Edit project** button.



This causes the following dialog to appear:



Editing ( ) a task on the live site is done through the dialog depicted below:

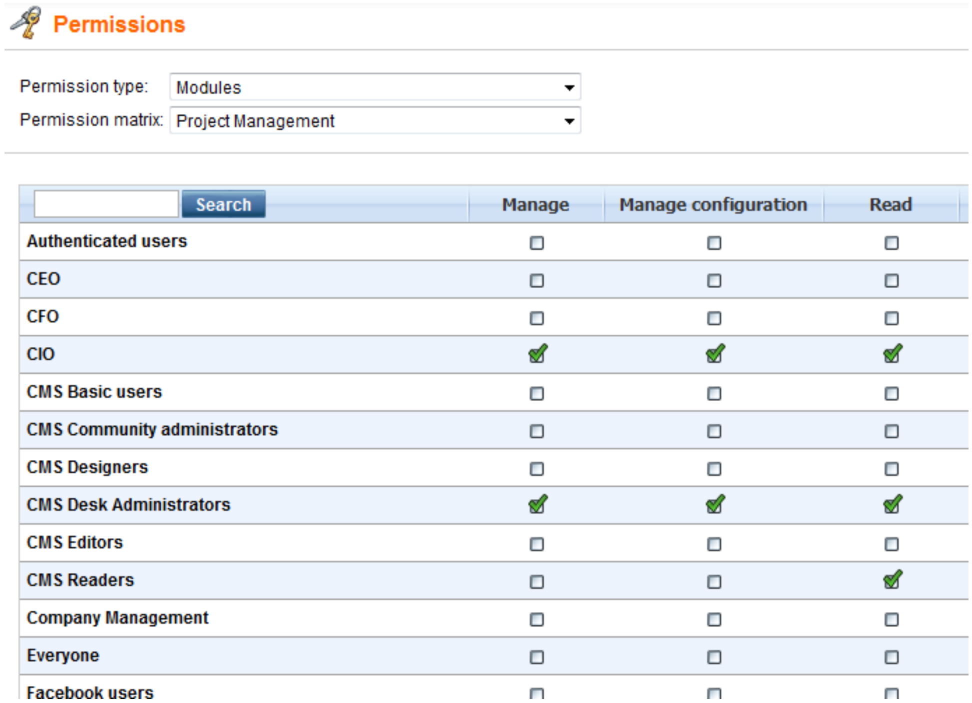


In addition to the task properties described in the [Managing projects and tasks] topic, the **Task URL** is shown, which can be used to link to the edited task.

Certain other web parts and widgets, such as **Tasks assigned to me**, may be used to edit displayed tasks in the same way.

8.32.5 Security

Permissions for administering the project management module can be set by going to **CMS Site Manager** (or **CMS Desk**) **-> Administration -> Permissions** and selecting the *Project Management* **Permission matrix**.


Permissions

Permission type: Modules

Permission matrix: Project Management

Search	Manage	Manage configuration	Read
Authenticated users	☐	☐	☐
CEO	☐	☐	☐
CFO	☐	☐	☐
CIO	☑	☑	☑
CMS Basic users	☐	☐	☐
CMS Community administrators	☐	☐	☐
CMS Designers	☐	☐	☐
CMS Desk Administrators	☑	☑	☑
CMS Editors	☐	☐	☐
CMS Readers	☐	☐	☑
Company Management	☐	☐	☐
Everyone	☐	☐	☐
Facebook users	☐	☐	☐

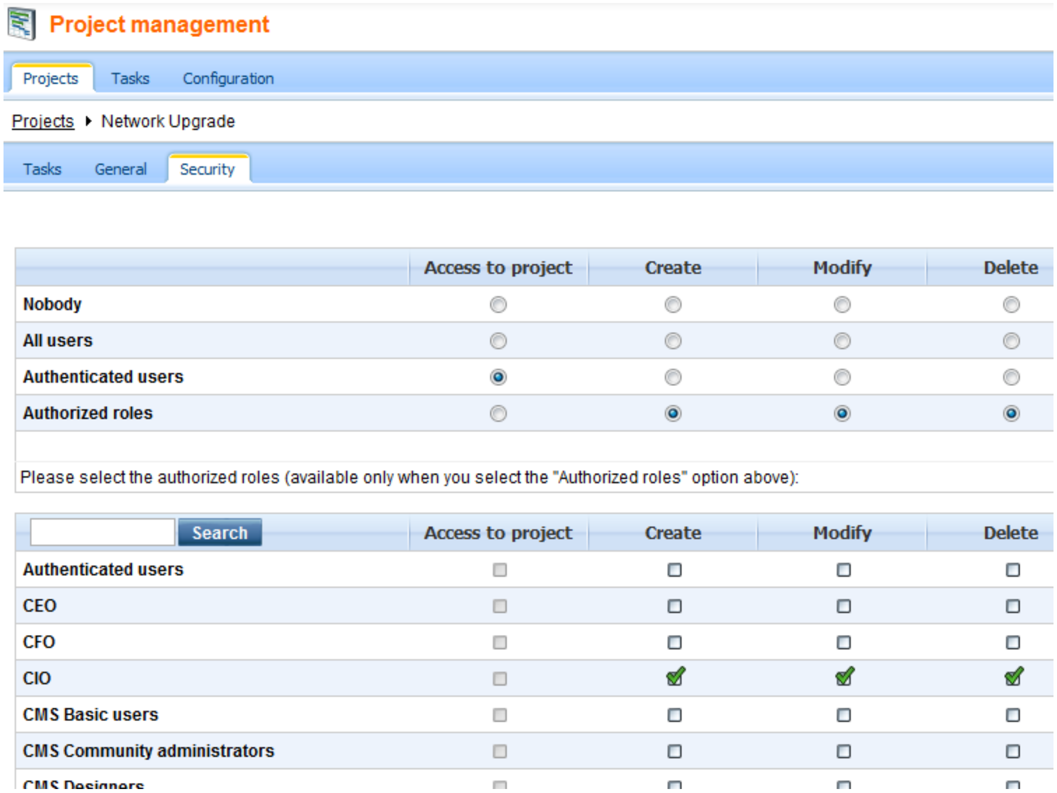
The following permissions can be assigned to members of the selected roles:

- **Manage** - members of the selected roles are allowed to create, modify and delete data in the project management interface, except for the *Configuration* tab
- **Manage configuration** - members of the selected roles are allowed to view and modify data, such as statuses and priorities, on the *Configuration* tab of the project management interface
- **Read** - members of the selected roles are allowed to view all data in the project management interface, except for the *Configuration* tab

To access and use the management interface for group projects, users must either have the **Read** and **Manage** permissions from the *Groups* **Permission matrix**, or belong to a group role that can manage the given group.

Project permissions

Permissions for individual projects can be configured on the **Security** tab when editing a project. This affects which users can view and manage the project and its tasks using project management web parts.



The following permissions are available:

- **Access to project** - users with this permission can view and access the project
- **Create** - users with this permission can create new tasks under the project
- **Modify** - users with this permission can modify the tasks under the project
- **Delete** - users with this permission can delete the tasks under the project

These permissions can be assigned to:

- **Nobody** - nobody can perform the action
- **All users** - all users can perform the action
- **Authenticated users** - only signed-in users can perform the action, i.e. anonymous public users cannot perform it
- **Group members** - only members of the group can perform the action; this option is only available for projects that belong under a group
- **Authorized roles** - only members that are assigned to the roles selected below can perform the action; for group projects, only the roles defined for the given group can be selected

Project security exceptions

1. The user who is set as the owner of a project automatically has full permissions for that project.

2. A task can be modified by users that it is assigned to or owned by regardless of the security settings of the project that the task belongs under.

3. Users who belong to roles that have the **Manage** permission for the project management module have full permissions for all projects that do not belong to a group.

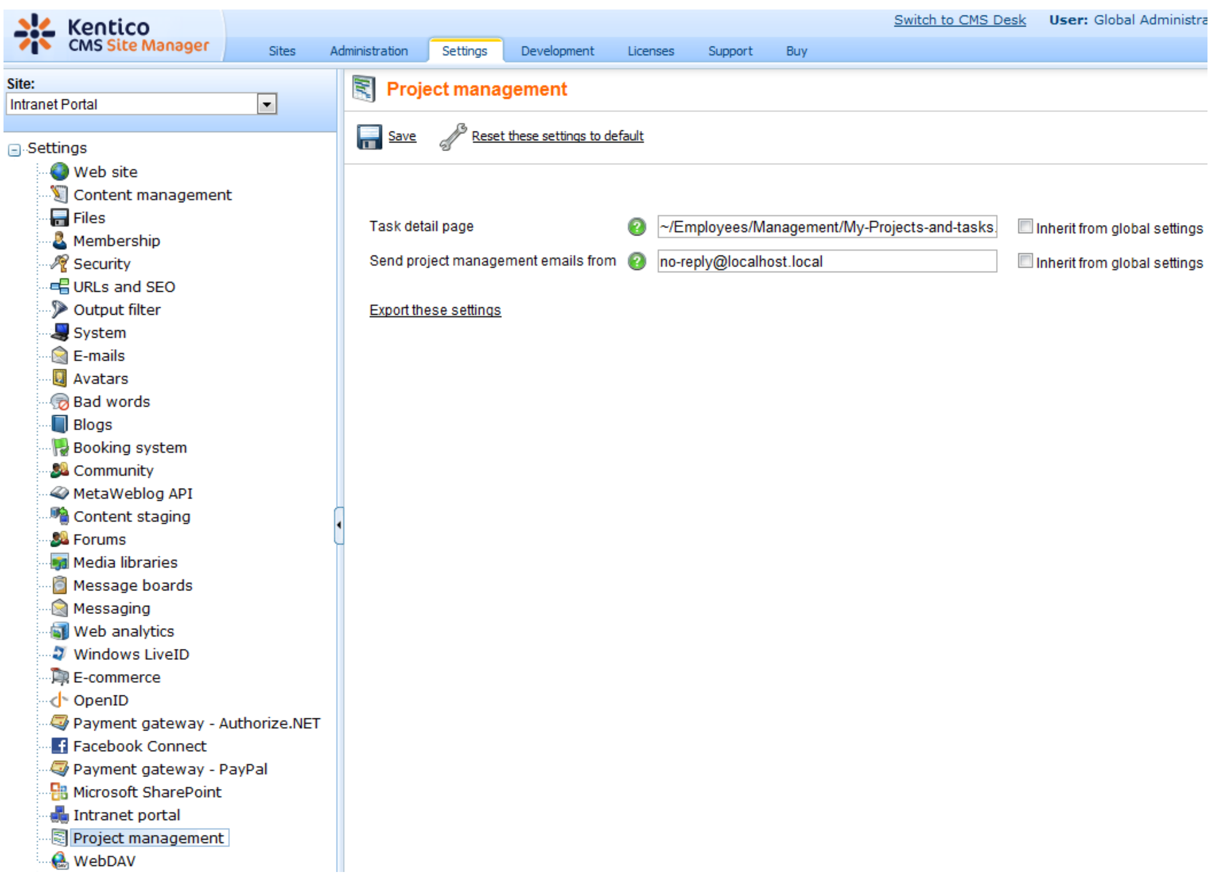
4. Users who belong to roles that can manage a group have full permissions for all projects under the given group.

5. The **Project list** web part has its own security properties that can be used to override the permissions set for the projects that it displays.

8.32.6 Settings

Additional settings of the project management module are located in **Site Manager -> Settings -> Project management**. The following can be configured:

- **Task detail page** - URL of a page that displays detailed information about the tasks assigned to users. This page should contain the *Tasks assigned to me* web part. The value of this setting is used for generating links to personal tasks in project management e-mail notifications and as the URL that is linked to by the *Task info panel* web part if its *Task detail page URL* property is empty. Sample value: *~/Employees/Management/My-Projects-and-tasks.aspx*
- **Send project management e-mails from** - Sets the e-mail address from which automatic project management notification e-mails are sent when tasks are created or modified.



8.33 Reporting

8.33.1 Overview

The Reporting module allows you to create internal reports to watch the activity in the Kentico CMS system and on websites, such as recently created documents, expired documents, website visits, user registration etc. The data the module uses is gathered from the database by using SQL queries and can be displayed in highly customizable reports including tables and graphs.

Report categories are used to group related reports together. See the Managing report categories topic for more details.

The following topics provide more details and examples on how reports can be created and used:

- Creating a new report
- Defining report parameters
- Saving a report
- Displaying a report on a web site

The Security topic describes how permissions can be set for the module.

The Reporting internals and API sub-chapter provides information about the database tables and classes used by the module and examples of how reports can be accessed using the API.

Reports are also used by the Web analytics module, which you can learn more about at Modules -> Web analytics.

Reports can be managed in **CMS Desk -> Tools -> Reporting**.

8.33.2 Managing report categories

All reports are organized into categories. It is recommended to keep reports that monitor related actions in one category. You can manage the categories in **CMS Desk -> Tools -> Reporting**.

Example:

The following steps describe how to create a sample report and how to display it on your website:

To start, you need to create a report category. Go to **CMS Desk -> Tools -> Reporting**, click **New report category** and enter the following values:

- **Category display name:** User reports
- **Category code name:** UserReports

Click **OK**.

Continued in the example section of the Creating a new report topic.

8.33.3 Creating a new report

8.33.3.1 Creating a new report

New reports can be created at **CMS Desk -> Tools -> Reporting -> Edit report category** under the **Reports** tab by using the **New report** link. Here you can also see a list of all the reports in this category and **Edit** (), **Delete** (), **Clone** () or **Export** () them if you wish.

When editing a report on its **General** tab, you can configure its properties, alter its layout using the built-in WYSIWYG editor and add file attachments. To retrieve and display data from the system in the report, some of the following objects must be added to the layout:

- Tables
- Graphs
- Values

Additionally, all types of macro expressions, as described in Appendix A - Macro expressions, can be used in the report layout.

The output of the report can be seen on the **View** tab.

Localizing strings in reports

If you need to create a single report in multiple languages, please use [Localization Expressions].

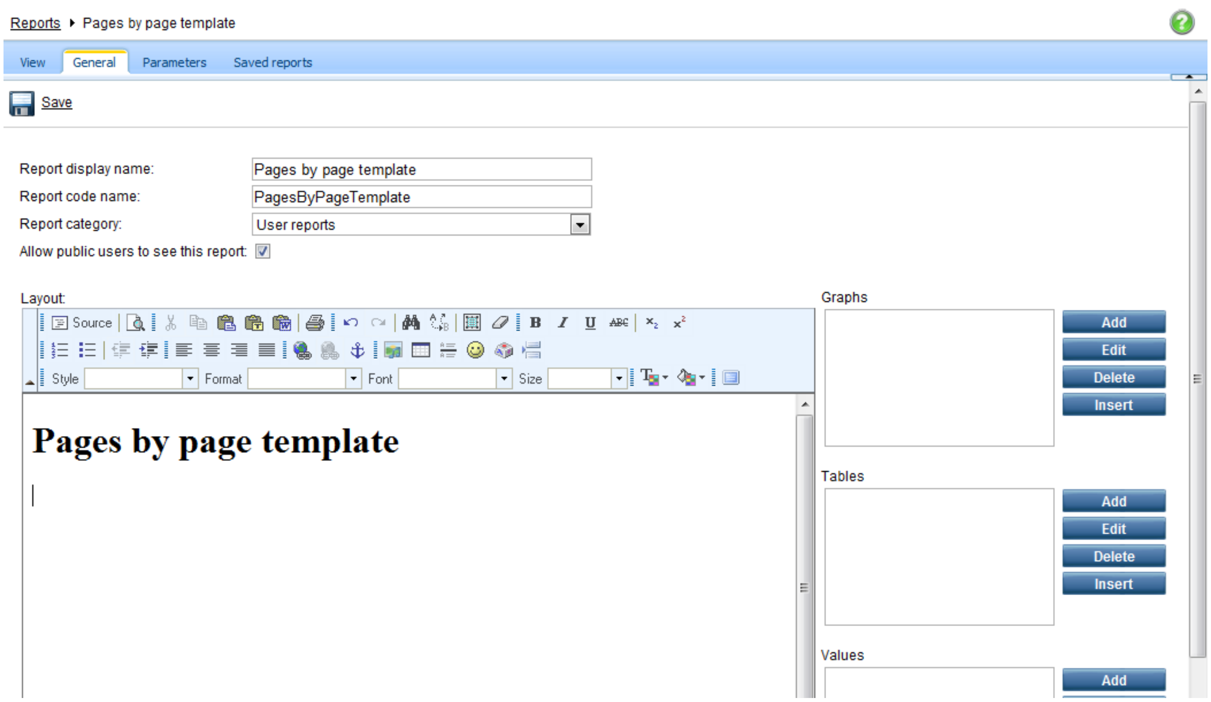
Example:

1. Click **New report** and enter the following values:

- **Report display name:** Pages by page template
- **Report code name:** PagesByPageTemplate

Click **OK**. Now you can edit the layout of the report and insert tables, graphs and values.

2. You will be redirected to the **General** tab of the report editing interface. Enter the following text in the **Layout** text area: *Pages by page template*. Select the text and use the WYSIWYG editor to set its **Format** to *Heading 1*.



Continued in the example section of the [Creating a new Table] topic.

8.33.3.2 Creating a new Table

A table can be placed into the layout of a report and can be used to display data retrieved from the Kentico CMS database by a query.

The following properties can be used to configure tables:

Display name	The name of the table shown in the user interface.
Code name	Name used in your code.
Query	Here you can add the SQL query used to retrieve data to be displayed by the table.
Is stored procedure	Indicates if the query is a stored procedure or not.
Skin ID	ID of the .NET skin (stored in the *.skin* files in *~/AppThemes/<theme name>*) which will be used for the table.
Enable paging	If enabled, paging will be enabled when the report table is displayed. The paging can be configured by the two properties below.
Page size	Number of table rows per page.
Page mode	Type of paging controls displayed below the table. The following options are available: • **Previous-next buttons** - displays buttons leading to the previous and next page • **Page numbers** - displays page numbers leading to the corresponding pages • **Previous-next-first-last buttons** - displays buttons leading to the first, last, previous and next page • **Page numbers-first-last buttons** - displays page numbers leading to the corresponding pages and buttons leading to the first and last page

Tables are entered into the report layout editor as an expression in the following format:

%%control:ReportTable?<report code name>.<table code name>%%

This is done automatically when the **Insert** button is used.

Writing queries for tables

The queries you write for tables are standard SQL queries that pull data from the Kentico CMS database. You can find the description of Kentico CMS database tables and views in **Kentico CMS Database Reference** that is a part of the standard installation.

For information about documents, you can use the *View_CMS_Tree_Joined* table that returns published versions of all documents.

Table column names

The table column names use the same names as the column names from the returned data set. If you need to use user friendly names, you can use the following syntax in the query:

```
SELECT PageTemplateDisplayName as [Template Name], ...
```

Example:

1. Click **Add** in the **Tables** section on the right. Enter the following values:

- **Display name:** Pages by Page Template
- **Code name:** PagesByPageTemplate
- **Query:**

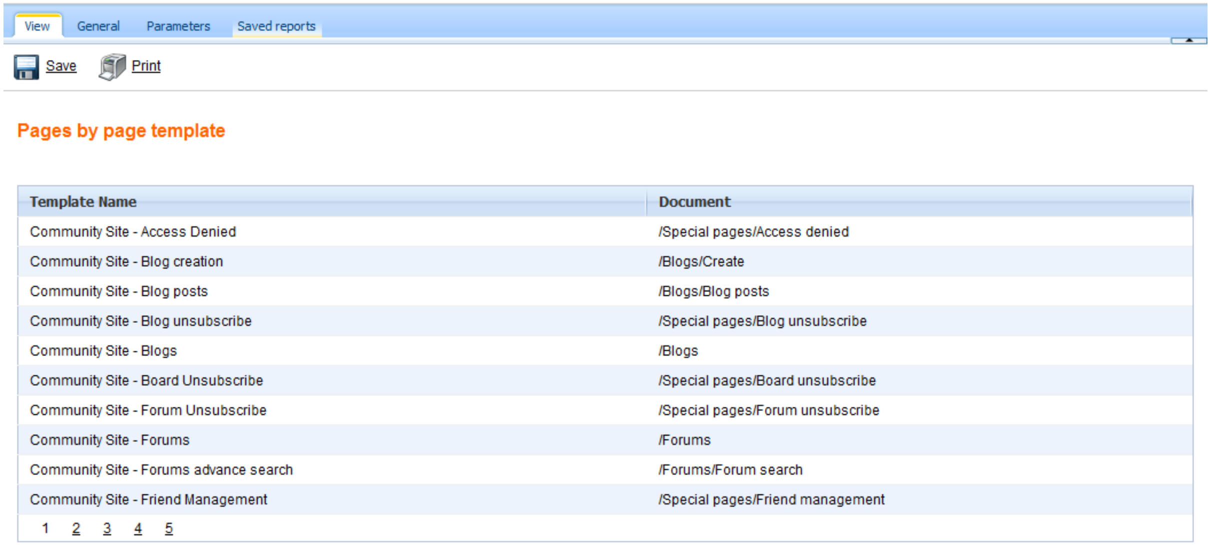
```
SELECT PageTemplateDisplayName as [Template Name], DocumentNamePath as [Document]
FROM view_CMS_Tree_Joined
LEFT JOIN cms_pagetemplate
ON cms_pagetemplate.pagetemplateid = view_CMS_Tree_Joined.DocumentPageTemplateID
WHERE pagetemplatedisplayname IS NOT NULL AND PageTemplateIsReusable = 1
ORDER BY PageTemplateDisplayName
```

- **Is stored procedure:** no
- **SkinID**: leave empty
- **Enable paging**: enabled
- **Page size**: 10
- **Page mode**: Page numbers

Click **OK**.

2. Now place the cursor in the layout editor on a new line under the title, choose the table in the **Tables** section and click **Insert**. A string like *%%control:ReportTable?PagesByPageTemplate.PagesByPageTemplate%%* will be added to the text area.

Click **Save** to save the changes and switch to the **View** tab. You will see a report like this:



Template Name	Document
Community Site - Access Denied	/Special pages/Access denied
Community Site - Blog creation	/Blogs/Create
Community Site - Blog posts	/Blogs/Blog posts
Community Site - Blog unsubscribe	/Special pages/Blog unsubscribe
Community Site - Blogs	/Blogs
Community Site - Board Unsubscribe	/Special pages/Board unsubscribe
Community Site - Forum Unsubscribe	/Special pages/Forum unsubscribe
Community Site - Forums	/Forums
Community Site - Forums advance search	/Forums/Forum search
Community Site - Friend Management	/Special pages/Friend management

Continued in the example section of the Creating a new Graph topic.

8.33.3.3 Creating a new Graph

A graph can be placed into the layout of a report and can be used to display data retrieved from the Kentico CMS database by a query.

The following properties can be used to configure graphs:

Display name	Display name of the graph
Code name	Code name of the graph
Query	Database query that extracts a dataset that will be displayed in the graph; it must return at least two columns - first one for categories, the other columns are used for values.
Is stored procedure	Determines if the specified query is a stored procedure.
Graph type	The following graph types are available: **Pie** - pie graph, accepts only one column for values, the value in the third column is used for setting the 'pie pieces' offset **Bar** - bar graph, accepts multiple values and displays them next to each other **Bar overlay** - bar graph, accepts multiple values and displays them behind each other with the lower values in the front **Bar stacked** - bar graph, accepts multiple values and displays them on top of each other **Bar percentage** - bar graph, accepts multiple values, displays one column for each category and divides it by percentage determined by

	the values **Line** - line graph, accepts multiple values and displays them as separate lines
Use vertical bars	If checked, the bars are displayed vertically instead of horizontally; used only for bar graphs
Reverse y-axis	Reverses the vertical axis; used only for bar and line graphs
Use ten powers	If large values are present in the graph, they are divided by appropriate ten powers and the division ratio is displayed with the y-axis title
Show grid	Displays a thin dotted line grid in the graph chart area; used only for bar and line graphs
Fill curves	Fills the space under the curves with a color; used only for line graphs
Smooth curves	Smooths the lines of the line graph
Title	Title of the graph
X axis title	Title of the horizontal axis of the graph
Y axis title	Title of the vertical axis of the graph
Width	Width of the graph image
Height	Height of the graph image
Legend position	Position of the legend in the graph
Title font	Determines the font properties of the graph title
Axis font	Determines the font properties of axis titles
X-axis angle	Determines the declination angle of X-axis descriptions; setting this parameter to 90 causes upright descriptions
Y-axis angle	Determines the declination angle of Y-axis descriptions
Scale min	Determines the minimum value that is required for an X axis category to be displayed
Scale max	Determines the maximum value that will be displayed on the Y axis of the graph
Use colors	Determines the colors used for graph items; accepts standard HTML color names or hexadecimal color codes divided by a semicolon; if it contains a lower number of colors than required, the remaining colors are generated automatically
Use symbols	Determines the symbols used for values in line graphs; accepts: *Circle, Diamond, HDash, Plus, Square, Star, Triangle, TriangleDown, VDash, XCross*; if *None* is entered, no symbols will be used
Graph gradient	Gradient of the graph background color; The **From** and **To** fields accept standard HTML color names or hexadecimal color codes, the **Angle** field sets gradient declination

Chart gradient	Color gradient of the graph chart; The **From** and **To** fields accept standard HTML color names or hexadecimal color codes, the **Angle** field sets gradient declination; used only for line and bar graphs
Item gradient	Color gradient of the items in the graph; The **From** field accepts standard HTML color names or hexadecimal color codes, the **Angle** field sets gradient declination; the **To** field is disabled here

Graphs are entered into the report layout editor as an expression in the following format:

%%control:ReportGraph?<report code name>.<graph code name>%%

This is done automatically when the **Insert** button is used.

Writing queries for pie charts

The queries for pie chart graphs must return two columns: the categories and their values. The graph automatically calculates the displayed size of the given category. If you return a constant value in the third column, this value will set the offset of pie pieces.

Writing queries for bar graphs

The queries for bar graphs must return at least two columns: the categories and their values. If you specify more than two columns, the additional columns will be displayed next to the second column value in the graph (**Bar**), in front of each other (**Bar overlay**), on top of each other (**Bar stacked**) or they will divide one column by percentage (**Bar percentage**).

Example:

1. Switch back to the **General** tab. Click **Add** in the **Graphs** section on the right. Enter the following values:

- **Display name:** Most Favorite Page Templates
- **Code name:** MostFavoritePageTemplates
- **Query:**

```
SELECT PageTemplateDisplayName AS [Template Name], count(PageTemplateDisplayName)
AS [Usage]
FROM view_CMS_Tree_Joined
LEFT JOIN cms_pagetemplate
ON cms_pagetemplate.pagetemplateid = view_CMS_Tree_Joined.DocumentPageTemplateID
WHERE pagetemplatedisplayname IS NOT NULL AND PageTemplateIsReusable = 1
GROUP BY PageTemplateDisplayName
ORDER BY count(PageTemplateDisplayName) DESC
```

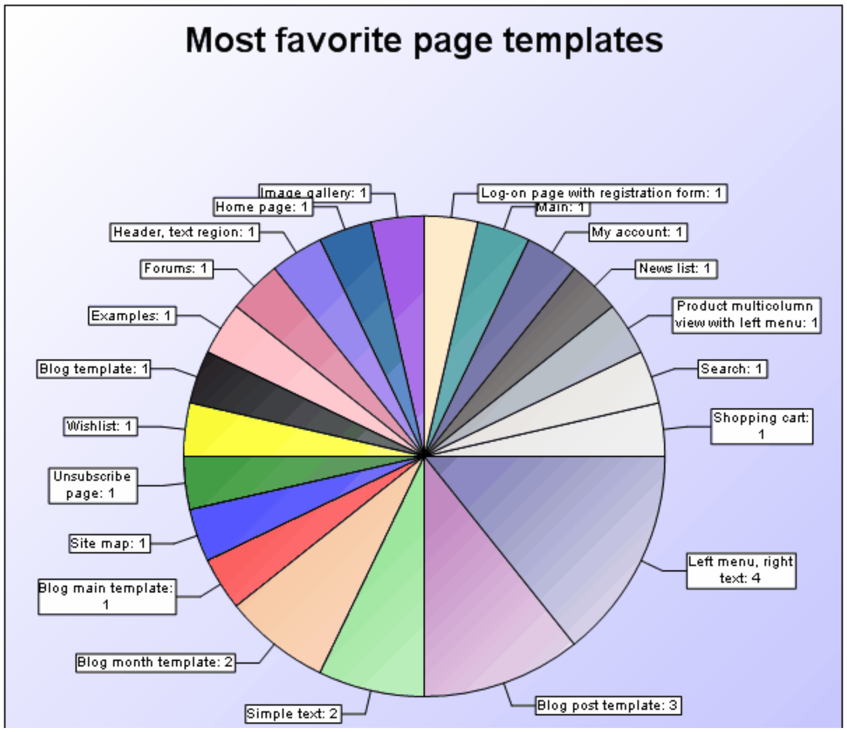
- **Is stored procedure:** no
- **Graph type:** Pie

- **Title:** Most favorite page templates
- **Width:** 600
- **Height:** 600
- **Legend position:** (No legend)

Click **OK**.

2. Now place the cursor in the layout editor on a new line under the table, select the new graph in the **Graphs** section and click **Insert**. A string like *%%control:ReportGraph?PagesByPageTemplate.MostFavoritePageTemplates%%* will be added to the text area.

Click **Save** to save the changes and switch to the **View** tab. You will see a graph like this in the report:



Continued in the example section of the [Creating a new Value] topic.

8.33.3.4 Creating a new Value

A value is an object that can be placed into the layout of a report, which can be used to display a single scalar value returned by a query in a specified string format.

The following properties of a value can be configured:

Display name	The name of the item in the list
Code name	Name used in your code
Query	Here you can add the SQL query used to retrieve data to be displayed by the value.

Is stored procedure	Indicates if the query is a stored procedure or not.
Formatting string	You can format the displayed value using the standard .NET expressions. Examples: • **{0}** - displays the value • **{0:F1}** - displays the value as a floating point number with one digit displayed after the decimal point You can find more details in .NET Framework documentation.

Values are entered into the report layout editor as an expression in the following format:

%%control:ReportValue?<report code name>.<value code name>%%

This is done automatically when the **Insert** button is used.

Writing queries for scalar value

The queries for scalar values may return any number of columns and rows, but the only value that will be displayed is the value in the first column of the first row of the result set.

Example:

1. Switch back to the **General** tab. Click **Add** in the **Values** section on the right. Enter the following values:

- **Display name:** Number of pages with page template
- **Code name:** PagesWithTemplate
- **Query:**

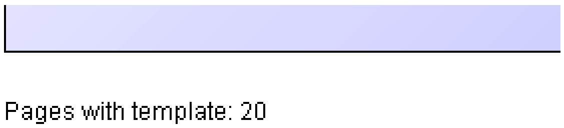
```
SELECT count(DocumentID)
FROM view_CMS_Tree_Joined
WHERE documentpagetemplateid IS NOT NULL
```

- **Is stored procedure**: no
- **Formatting string**: Pages with template: {0}

Click **OK**.

2. Place the cursor in the layout editor under the graph, choose the new value in the **Values** section and click **Insert**. A string like *%%control:ReportValue?PagesByPageTemplate.PagesWithTemplate%%* will be added to the text area.

Click **Save** to save the changes and switch to the **View** tab. You will see a text like this:



Continued in the example section of the Defining report parameters topic.

8.33.4 Defining report parameters

Reports may be filtered using parameters. You can define custom parameters on the **Parameters** tab of the **Report properties** dialog.

Context Parameters

In your queries, you can use parameters that provide information about the current context when the report is viewed, such as current user, current site, etc. Here's a list of all available context variables:

- @CMSContextCurrentUserID
- @CMSContextCurrentUserName
- @CMSContextCurrentUserDisplayName
- @CMSContextCurrentSiteID
- @CMSContextCurrentSiteName
- @CMSContextCurrentSiteDisplayName
- @CMSContextCurrentDomain
- @CMSContextCurrentTime
- @CMSContextCurrentURL
- @CMSContextCurrentNodeID
- @CMSContextCurrentCulture
- @CMSContextCurrentDocumentID
- @CMSContextCurrentAliasPath
- @CMSContextCurrentDocumentName
- @CMSContextCurrentDocumentNamePath

For example, if you want to display a list of all expired documents of the current website, you can use a query like this:

```
SELECT DocumentNamePath as [Document path]
FROM View_CMS_Tree_Joined
WHERE documentpublishto < @CMSContextCurrentTime and nodesiteid =
@CMSContextCurrentSiteID
```

Displaying Parameter Values in the Report

If you need to display the parameter values in the report, you can place the following macro expression in the report text:

{%parametername%}

For example:

List of documents expired on or before {%CMSContextCurrentTime%}

displays:

List of documents expired on or before 8/12/2007 12:06:49 PM

You can use this syntax for both custom report parameters and context parameters.

Example:

1. Switch to the **Parameters** tab, click **New attribute** ( ) and enter the following values:

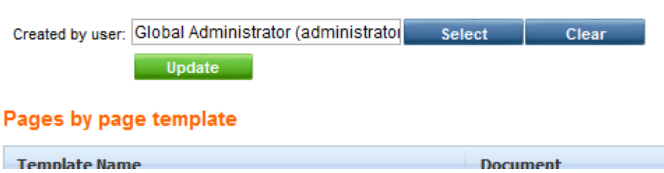
- **Attribute name:** UserID
- **Attribute type:** Integer number
- **Attribute default value:** 53
- **Field caption:** Created by user
- **Field type:** User Selector

Click **OK**.

2. Now we need to add this parameter to our queries. For the purposes of this example, we will modify only the table query. Switch to the **General** tab, select the **Pages by page template** table and click edit. Now modify the table SQL query like this:

```
SELECT PageTemplateDisplayName as [Template Name], DocumentNamePath as [Document]
FROM view_CMS_Tree_Joined
LEFT JOIN cms_pagetemplate
ON cms_pagetemplate.pagetemplateid = view_CMS_Tree_Joined.DocumentPageTemplateID
WHERE pagetemplatedisplayname IS NOT NULL AND DocumentCreatedByUserID = @UserID
ORDER BY PageTemplateDisplayName
```

As you can see we added the parameter to the WHERE condition of the query. All parameters that you define can be used in the query using the @<parametername> expression. Click **OK** and go to the **View** tab. You will see that the report now has a filter like this:



The table now only displays template names of documents that were created by the user specified in the filter.

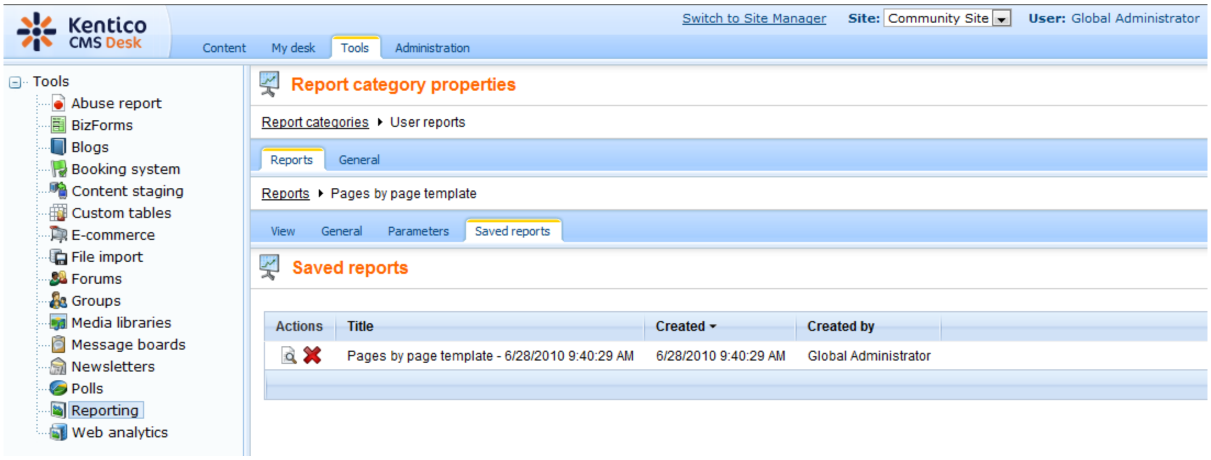
Continued in the example section of the Saving a report topic.

8.33.5 Saving a report

When you view some report on the **View** tab of the **Report properties** dialog, you can save it into the report history using the **Save** button. You can view the saved reports on the **Saved reports** tab. When the data changes, the saved reports are not affected.

Example:

Click the **Save** button on the **View** tab. Now switch to the **Saved reports** tab. Here you can see a list of saved reports, it should look something like this:



Concluded in the example section of the [Displaying a report on a web site] topic.

8.33.6 Displaying a report on a website

If you want to display some report on the website or include it in some custom page in the CMS administration interface, you can use the **Reporting -> Report** web part. All you need to configure are the following properties:

- **Report name** - select the required report
- **Display filter** - indicates if a filter should be displayed on the page (if the report has some parameters specified)

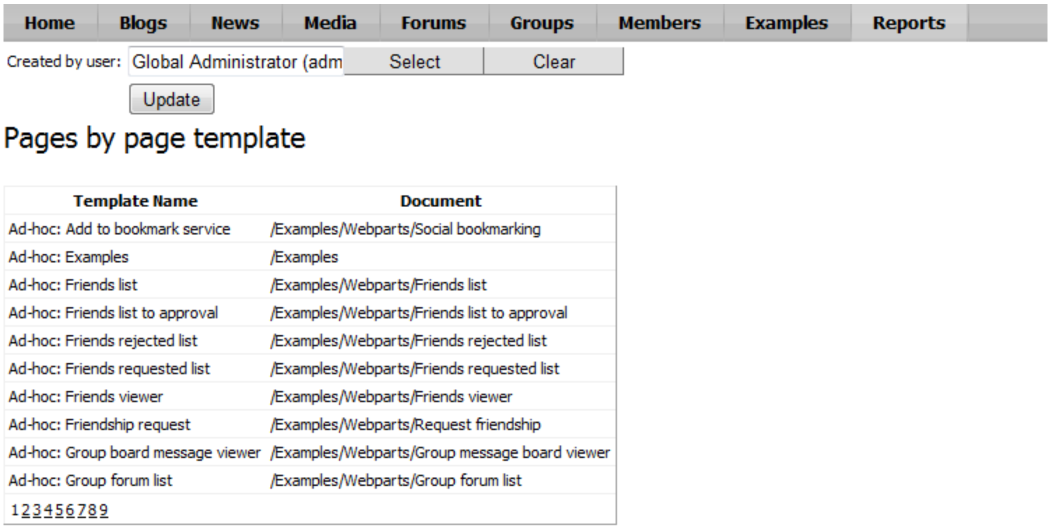
Example:

1. To display this report on your website, switch to the **Content** tab of CMS Desk. Select the root of your website and create a new **Page (menu item)** type document. Name it *Reports*, select **Create a blank page,** choose the **Simple** layout and click **Save**.

2. Now add the **Reporting/Report** web part to the page's web part zone and set its properties to the following:

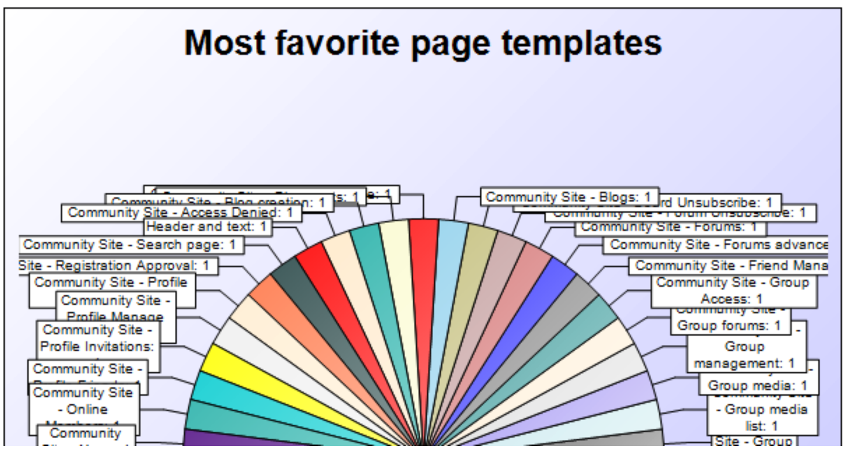
- **Report name** - select *Pages by page template*
- **Display filter** - enabled

3. Now go to **Live site** mode and you should see something similar to the following:



Template Name	Document
Ad-hoc: Add to bookmark service	/Examples/Webparts/Social bookmarking
Ad-hoc: Examples	/Examples
Ad-hoc: Friends list	/Examples/Webparts/Friends list
Ad-hoc: Friends list to approval	/Examples/Webparts/Friends list to approval
Ad-hoc: Friends rejected list	/Examples/Webparts/Friends rejected list
Ad-hoc: Friends requested list	/Examples/Webparts/Friends requested list
Ad-hoc: Friends viewer	/Examples/Webparts/Friends viewer
Ad-hoc: Friendship request	/Examples/Webparts/Request friendship
Ad-hoc: Group board message viewer	/Examples/Webparts/Group message board viewer
Ad-hoc: Group forum list	/Examples/Webparts/Group forum list

1 2 3 4 5 6 7 8 9



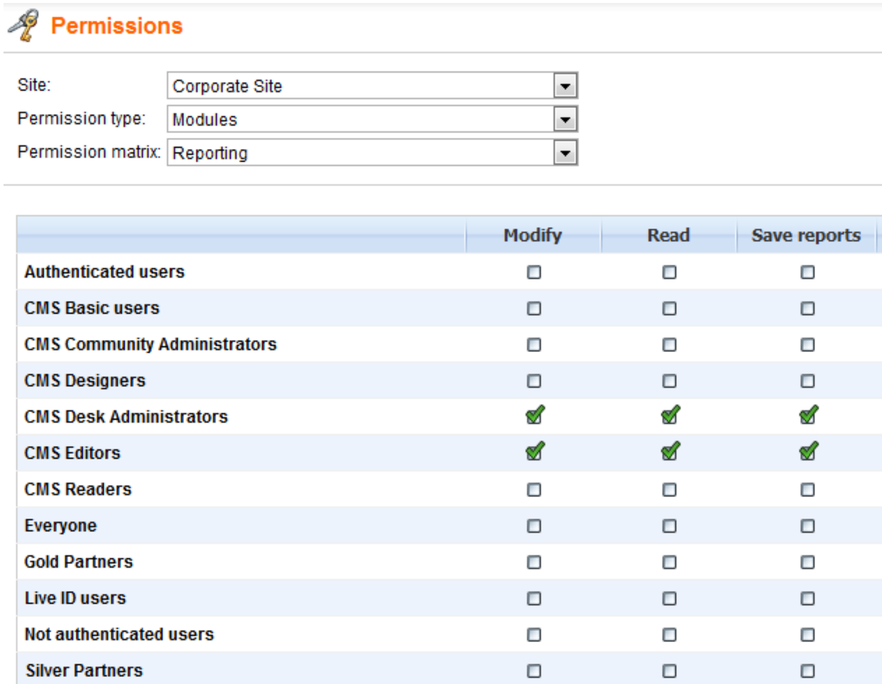
As you can see, the report is displayed just like on the **View** tab of the **Report Properties** dialog and the parameter filter can be used by site visitors.

8.33.7 Security

You can configure access to the Reporting module in **CMS Desk -> Administration -> Permissions**. Choose the permission matrix for **Modules** -> **Reporting.**

You can assign roles with the following permissions:

- **Modify -** allows users to create, modify and delete reports
- **Read** - allows users to read existing reports
- **Save reports** - allows users to save reports



Making reports available on the live site

The **Allow public users to see this report** property, which can be found on the **General** tab when **Editing** ( ) a report, indicates if the report can be displayed on the live site to non-authenticated (public) users. If it's set to false, the report and its graphs will be hidden from public users.

8.33.8 Reporting internals and API

8.33.8.1 Database tables and API classes

The Reporting module uses the following database tables:

- **Reporting_Report** - contains reports and their settings and layout
- **Reporting_ReportCategory** - contains report categories
- **Reporting_ReportGraph** - contains report graphs and their settings
- **Reporting_ReportTable** - contains report tables and their settings
- **Reporting_ReportValue** - contains report values and their settings
- **Reporting_SavedGraph** - contains graphs saved with saved reports (in binary format)
- **Reporting_SavedReport** - contains saved reports

The Reporting API is provided by the following **CMS.Reporting** namespace classes:

- **ReportCategoryInfo, ReportCategoryInfoProvider** - these classes provide functionality for managing report categories
- **ReportGraphInfo, ReportGraphInfoProvider** - these classes provide functionality for managing report graphs
- **ReportInfo, ReportInfoProvider** - these classes provide functionality for managing reports
- **ReportTableInfo, ReportTableInfoProvider** - these classes provide functionality for managing report tables

- **ReportValueInfo, ReportValueInfoProvider** - these classes provide functionality for managing report values
- **SavedGraphInfo, SavedGraphInfoProvider** - these classes provide functionality for managing saved graphs (for saved reports)
- **SavedReportInfo, SavedReportInfoProvider** - these classes provide functionality for managing saved reports

The following topics show examples of how these classes can be used:

- Getting report data
- Managing reports and their categories
- Creating graphs, tables and values
- Editing the layout of a report
- Saving the current status of the report
- Getting HTML code of a saved report

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.33.8.2 Getting report data

The following sample code shows how you can get a report category as a *ReportCategoryInfo* object and a report as a *ReportInfo* object:

[C#]

```
using CMS.Reporting;

...

        // Get report category object by category code name
        ReportCategoryInfo rci = ReportCategoryInfoProvider.GetReportCategoryInfo("W

        // Get report category object by category ID
        ReportCategoryInfo rci2 = ReportCategoryInfoProvider.GetReportCategoryInfo(1

        // Get report category object by report ID
        ReportInfo ri = ReportInfoProvider.GetReportInfo(1);

        // Get report category object by report code name
        ReportInfo ri2 = ReportInfoProvider.GetReportInfo("TestReport");
```

The following sample code shows how you can get a DataSet containing report categories or reports from the system:

[C#]

```
using System.Data;
using CMS.Reporting;

...

        string where = "CategoryID > 10";
        string orderby = "CategoryDisplayName";
        int topN = 10;
        string selectedColumns = "CategoryID, CategoryDisplayName";

        // Get dataset of report categories according to the given parameters
        DataSet ds = ReportCategoryInfoProvider.GetCategories(where, orderby);

        DataSet ds2 = ReportCategoryInfoProvider.GetCategories(where, orderby, topN,


        string where2 = "ReportCategoryID = 12";

        // Get DataSet of reports with a specified where condition
        DataSet ds3 = ReportInfoProvider.GetReports(where2);
```

8.33.8.3 Managing reports and their categories

The following sample code shows how you can create a new report category and add it to the system:

[C#]

```
using CMS.Reporting;

...

        // Create new report category object
        ReportCategoryInfo rci = new ReportCategoryInfo();

        // Set properties
        rci.CategoryDisplayName = "Test category";
        rci.CategoryCodeName = "TestCategory";

        // Save to database
        ReportCategoryInfoProvider.SetReportCategoryInfo(rci);
```

The following sample code shows how you can create a new report under an existing report category and add it to the system:

[C#]

```
using CMS.Reporting;

...
```

```
        // Get parent category for the new report
        ReportCategoryInfo parent = ReportCategoryInfoProvider.GetReportCategoryInfo

        // If parent exists
        if (parent != null)
        {
            // Create new report object
            ReportInfo ri = new ReportInfo();

            // Set properties
            ri.ReportDisplayName = "Test report";
            ri.ReportName = "TestReport";
            ri.ReportCategoryID = parent.CategoryID;
            ri.ReportAccess = ReportAccessEnum.All;
            ri.ReportLayout = "";
            ri.ReportParameters = "";

            // Save to database
            ReportInfoProvider.SetReportInfo(ri);
        }
```

The following sample code shows how you can delete an existing report:

[C#]

```
using CMS.Reporting;

...

        // Get report object by report code name
        ReportInfo ri = ReportInfoProvider.GetReportInfo("TestReport");

        // If report exists
        if (ri != null)
        {
            ReportInfoProvider.DeleteReportInfo(ri);
        }
```

8.33.8.4 Creating graphs, tables and values

The following sample code shows how you can create a new graph under an existing report:

[C#]

```
using CMS.Reporting;

...

        // Get report object by report code name
```

```
        ReportInfo ri = ReportInfoProvider.GetReportInfo("TestReport");

        // If report exists
        if (ri != null)
        {
            // Create new graph object
            ReportGraphInfo rgi = new ReportGraphInfo();

            // Set properties
            rgi.GraphDisplayName = "Test graph";
            rgi.GraphName = "TestGraph";
            rgi.GraphQuery = "SELECT TOP 10 DocumentName, DocumentID FROM CMS_Docume
            rgi.GraphReportID = ri.ReportID;
            rgi.GraphQueryIsStoredProcedure = false;

            rgi.GraphType = "bar";

            // Save to database
            ReportGraphInfoProvider.SetReportGraphInfo(rgi);

        }
```

The following sample code shows how you can create a new table under an existing report:

[C#]

```
using CMS.Reporting;

...

        // Get report object by report code name
        ReportInfo ri = ReportInfoProvider.GetReportInfo("TestReport");

        // If report exists
        if (ri != null)
        {
            // Create new table object
            ReportTableInfo rti = new ReportTableInfo();

            // Set properties
            rti.TableDisplayName = "Test table";
            rti.TableName = "TestTable";
            rti.TableQuery = "SELECT TOP 10 DocumentName, DocumentID FROM CMS_Docume
            rti.TableReportID = ri.ReportID;
            rti.TableQueryIsStoredProcedure = false;

            // Save to database
            ReportTableInfoProvider.SetReportTableInfo(rti);

        }
```

The following sample code shows how you can create a new value under an existing report:

[C#]

```
using CMS.Reporting;

...

        // Get report object by report code name
        ReportInfo ri = ReportInfoProvider.GetReportInfo("TestReport");

        // If report exists
        if (ri != null)
        {
            // Create new value object
            ReportValueInfo rvi = new ReportValueInfo();

            // Set properties
            rvi.ValueDisplayName = "Test value";
            rvi.ValueName = "TestValue";
            rvi.ValueQuery = "SELECT COUNT(DocumentName) FROM CMS_Document";
            rvi.ValueReportID = ri.ReportID;
            rvi.ValueQueryIsStoredProcedure = false;

            // Save to database
            ReportValueInfoProvider.SetReportValueInfo(rvi);

        }
```

8.33.8.5 Editing the layout of a report

The following sample code shows how you can insert items into the layout of a report:

[C#]

```
using CMS.Reporting;

...

        // Get report object by report code name
        ReportInfo ri = ReportInfoProvider.GetReportInfo("TestReport");

        // If report exists
        if (ri != null)
        {

            // Get graph object by code name
            ReportGraphInfo rgi = ReportGraphInfoProvider.GetReportGraphInfo("TestGr
            if (rgi != null)
            {
                ri.ReportLayout += "<br/>%%control:ReportGraph?" + ri.
ReportName + "." + rgi.GraphName + "%%<br/>";
```

```
            }

            // Get table object by code name
            ReportTableInfo rti = ReportTableInfoProvider.GetReportTableInfo("TestTa
            if (rti != null)
            {
                ri.ReportLayout += "<br/>%%control:ReportTable?" + ri.
ReportName + "." + rti.TableName + "%%<br/>";
            }

            // Get value object by code name
            ReportValueInfo rvi = ReportValueInfoProvider.GetReportValueInfo("TestVa
            if (rvi != null)
            {
                ri.ReportLayout += "<br/>%%control:ReportValue?" + ri.
ReportName + "." + rvi.ValueName + "%%<br/>";
            }

            // Save to database
            ReportInfoProvider.SetReportInfo(ri);
        }
```

8.33.8.6 Saving the current status of the report

The following sample code shows you how to execute a report and save its status into a saved report. You will need to create an ASCX user control where you place the **CMSReporting\DisplayReport.ascx** user control with ID set to *displayReportControl*. The code behind will look like this:

[C#]

```
using System.Data;

protected void Page_Load(object sender, EventArgs e)
{

        // Create a new data table with parameters
        DataTable dtp = new DataTable();
        dtp.Columns.Add("ModifiedSince", typeof(DateTime));

        object[] parameters = new object[1];
        parameters[0] = new DateTime(2010, 1, 1);

        dtp.Rows.Add(parameters);

        // Disable loading parameters from basic form
        displayReportControl.LoadFormParameters = false;

        // Choose a report to save by code name
        displayReportControl.ReportName = "RecentlyModifiedDocuments";

        // Set custom parameters
        displayReportControl.ReportParameters = dtp.Rows[0];
```

```
        // Reload data with new parameters
        displayReportControl.ReloadData(false);

        // SaveReport() returns 0 if some problem occurred or
        // SavedReportID of the successfully saved report
        displayReportControl.SaveReport();
}

// This override must be added if the report contains a table to avoid getting the "
// form tag with runat=server" error
public override void VerifyRenderingInServerForm(Control control)
{
        return;
}
```

8.33.8.7 Getting HTML code of a saved report

The following code sample shows how to read the HTML code of a saved report and display it on a page. It uses a **Literal** control with ID set to *ltrReportHTML* to do so:

[C#]

```
using CMS.Reporting;

...

        // Saved report id
        int mSavedReportId = 10;

        // Saved report info object
        SavedReportInfo sri = null;

        // Get saved report info by id
        sri = SavedReportInfoProvider.GetSavedReportInfo(mSavedReportId);

        // If saved report exists
        if (sri != null)
        {
            // Set saved report HTML to the literal control text
            ltrReportHTML.Text = sri.SavedReportHTML;
        }
```

8.34 Syndication (RSS, Atom, XML)

8.34.1 Overview

The Syndication module allows you to create syndication feeds of your site's content. The whole functionality is provided by a set of web parts stored under the **Syndication** web part category. These web parts generate feeds which conform to the RSS, Atom or general XML format. The name of the feed format generated by each web part is included in the name of the web part.

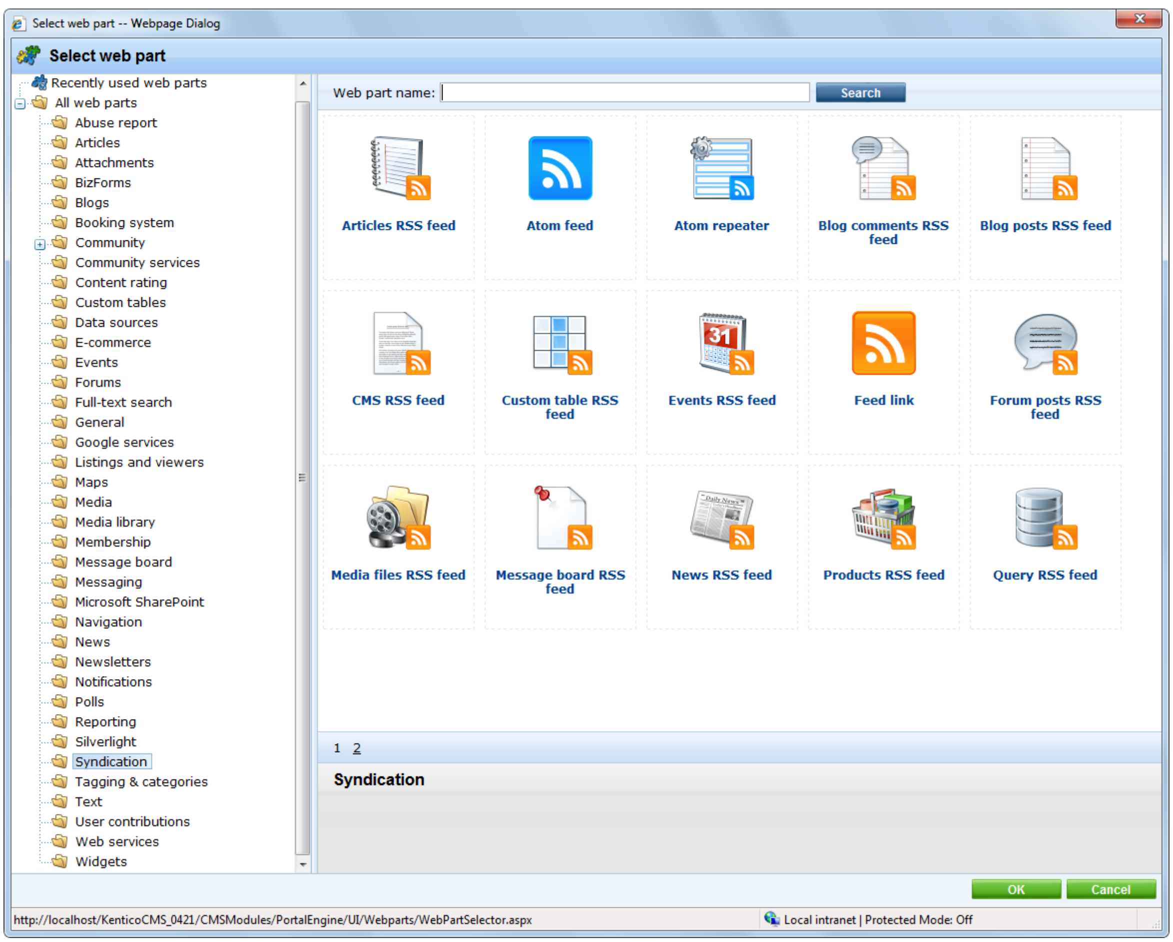
Detailed overview of the web parts can be found in the Syndication web parts and widgets topic. Use of transformations is essential with these web parts. Information related to use of transformations in syndication web parts can be found in the Syndication transformations topic.

We have also prepared three examples of typical usage:

- Example 1: Creating a **document RSS feed** using the *CMS RSS feed* web part.
- Example 2: Creating an **object RSS feed** using the *RSS feed* web part and a *data source*.
- Example 3: Creating a **dedicated page with an RSS feed** using the *RSS repeater* web part and a *data source*.

Live usage examples of these web parts can be found on the sample **Corporate Site**, under the **/ Examples/Web-parts/Syndication/** node.

All specific web part properties are explained in Kentico CMS Web Parts Reference or after clicking the **Documentation** ( ) link in the top right corner of the *Web part properties* dialog.



Creating RSS feeds manually

Due to backward compatibility with Kentico CMS versions prior to 5.5, it is still possible to create a page with an RSS feed manually. Even though it is not recommended to take this approach, you can learn more about it in this chapter.

8.34.2 Syndication web parts and widgets

Syndication web parts are stored under the **Syndication** category in the web part catalog. There is a large number of them, but most of them are similar and can be grouped into several categories:

From the functional point of view, there are two basic types of syndication web parts - repeaters and feeds:

- **Syndication feeds** show the feed icon with an optional text. The icon and text are clickable and when clicked, a querystring parameter with the feed ID is appended to its URL and the page renders the feed.

- **Syndication repeaters** change the page on which they are placed to a feed. If you place one of the syndication repeaters on a page, it changes the page response type to *application/xml*, transforms retrieved data to the required feed format and uses the transformed data as the content of the feed. This means that the page is no longer a standard HTML page - it becomes a syndication feed.

The web parts can also be divided in terms of what data they are used for. From this perspective, you can divide them as follows:

Basic feed web parts

These are universal web parts which can create a feed of data provided by a connected data source. Without a data source, the web parts are not functional. The data provided by the data source can be of any type, it just needs to be handled properly by the used transformation.

- **XML repeater**
- **RSS repeater**
- **Atom repeater**
- **RSS feed**
- **Atom feed**

Document feed web parts

These web parts are based on the **RSS feed** web part, but have a built-in data source for the **document types** in their names. This means that you don't need a connected data source - all you need is included in a single web part. They are a kind of "all-in-one" solutions for frequently used document types.

- **CMS RSS Feed**
- **Articles RSS Feed**
- **Blog posts RSS Feed**
- **Events RSS Feed**
- **News RSS Feed**

The **CMS RSS Feed** can even be used for **documents of any type**, which makes it a universal web part for any document feed.

Object feed web parts

Similarly to object feeds described above, object feeds also have a built-in data source, letting you create feeds from **Kentico CMS objects** by adding just a single "all-in-one" web part.

- **Blog comments RSS feed**
- **Custom tables RSS feed**
- **Media files RSS feed**
- **Message board RSS feed**
- **Products RSS feed**
- **Forum posts RSS feed**

Other syndication web parts

Custom feeds can be created using the following two web parts:

- **Query RSS feed -** generates a feed based on a custom database query and transformation
- **Web service RSS feed -** transforms a dataset provided by a web service into an RSS feed

The last web part does not crate any feed, but can be used as a link to a feed:

- **Feed link -** displays the RSS icon with a link leading to a URL set in its web part properties; typically used for links to feeds created by syndication repeaters

Syndication widgets

The default installation of Kentico CMS contains a set of widgets which are derived from the web parts listed above. They provide the same functionality and look, while only a limited set of properties can be configured. The following widgets are available:

- **Articles RSS Feed**
- **Blog comments RSS feed**
- **Blog posts RSS Feed**
- **Events RSS Feed**
- **Forum posts RSS feed**
- **Media files RSS feed**
- **News RSS Feed**
- **Products RSS feed**

Further information

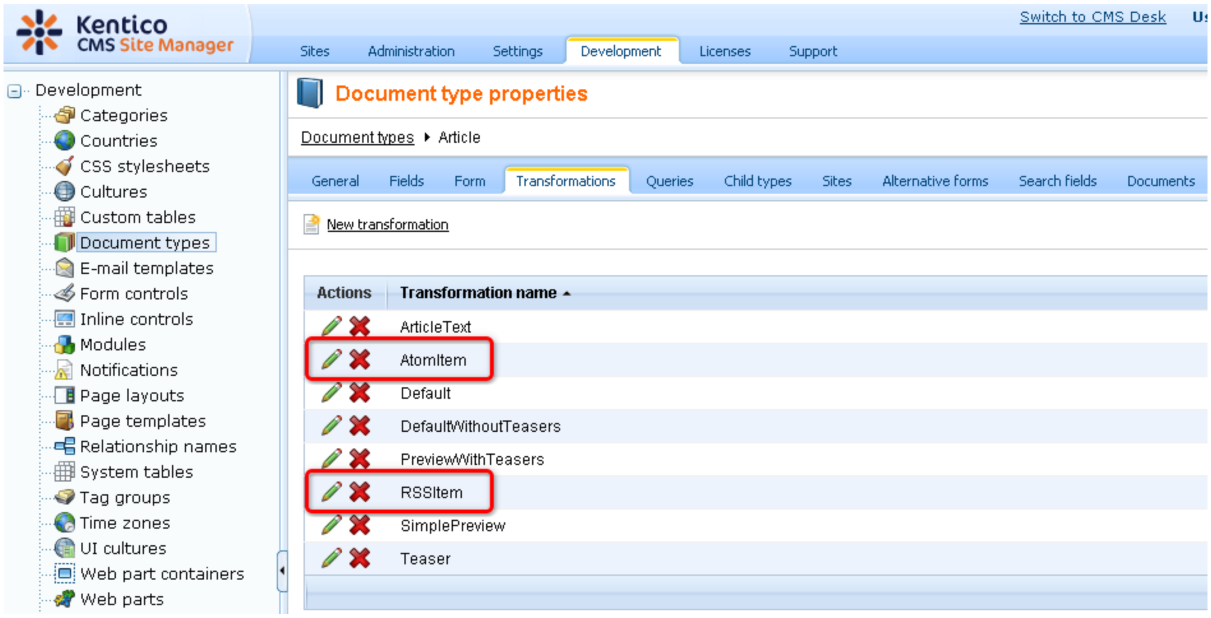
All specific properties of these web parts or widgets are explained in Kentico CMS Web Parts Reference or after clicking the **Documentation** ( ) link in the top right corner of the *Web part (widget) properties* dialog.

8.34.3 Syndication transformations

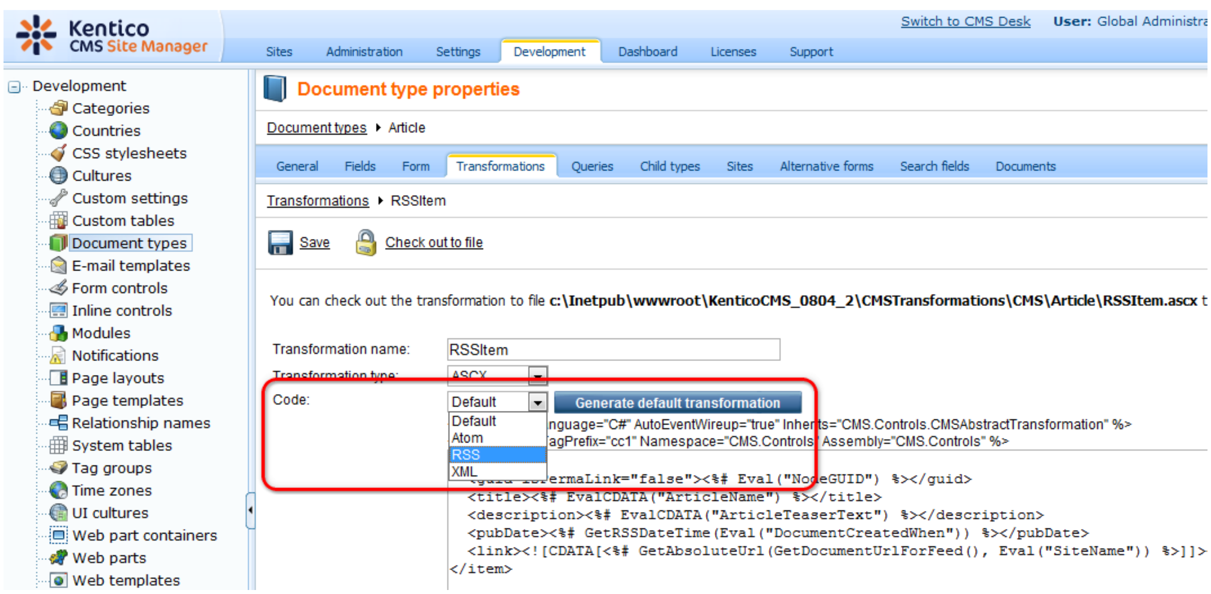
Use of transformations is of the highest importance when creating feeds, as all syndication web parts render the feeds using them. They need to be set in the **Transformation** property of each feed web part.

Default transformations

All document types in Kentico CMS have the **RSSItem** and **AtomItem** transformations. When you create a new document type, these transformations are created for it automatically.

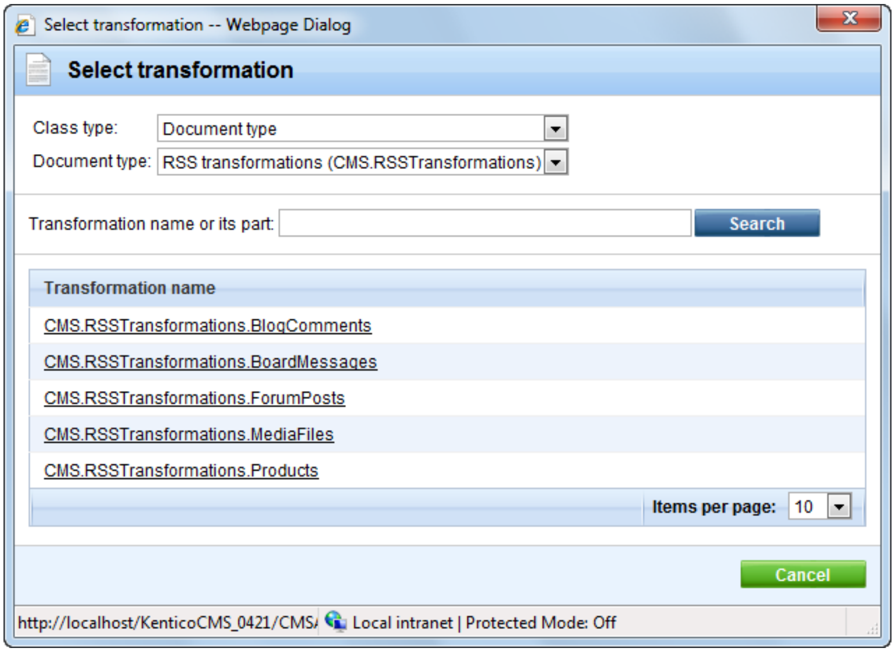


The default RSS, Atom and XML transformations can also be generated when editing or creating transformations of a document type or a custom table. To do this, you only need to select the required format from the **Code** drop-down list and click the **Generate default transformation** button.



RSS transformations document type

Other feed transformations are found in a special document type called **RSS transformations (CMS. RSSTransformations)**. This is a special document type used just to store transformations for **Kentico CMS objects**. Transformations for blog comments, board messages, forum posts, media library files and products are contained in the document type.



8.34.4 Usage examples

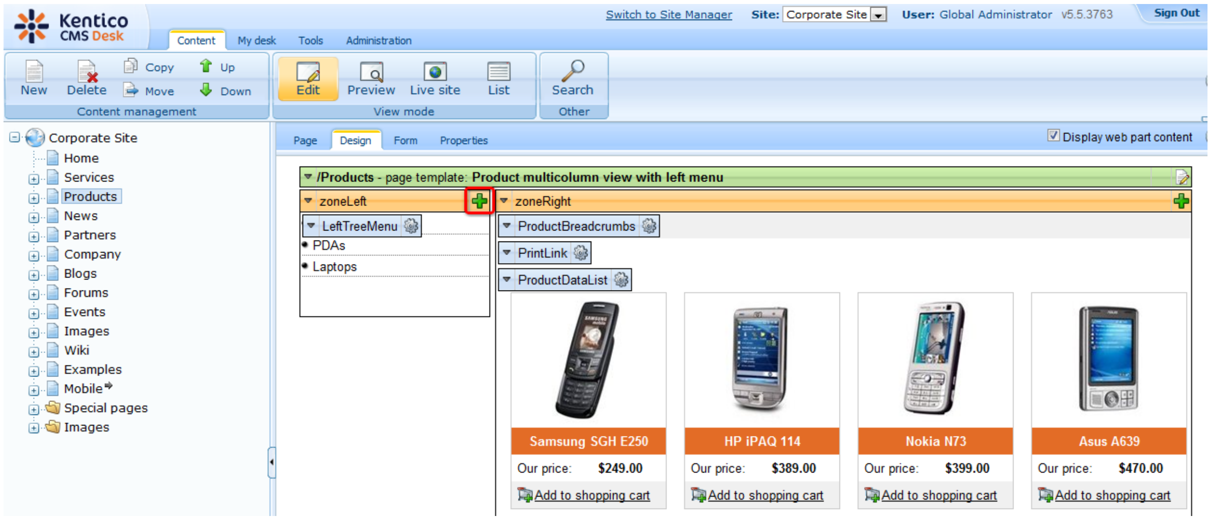
8.34.4.1 CMS RSS feed

The **CMS RSS feed** web part can be used to easily create RSS feeds from documents in the content tree. It has a **built-in Documents data source**, which means that the web part can work separately without any connected data source and that it can be used to create feeds from documents of any type.

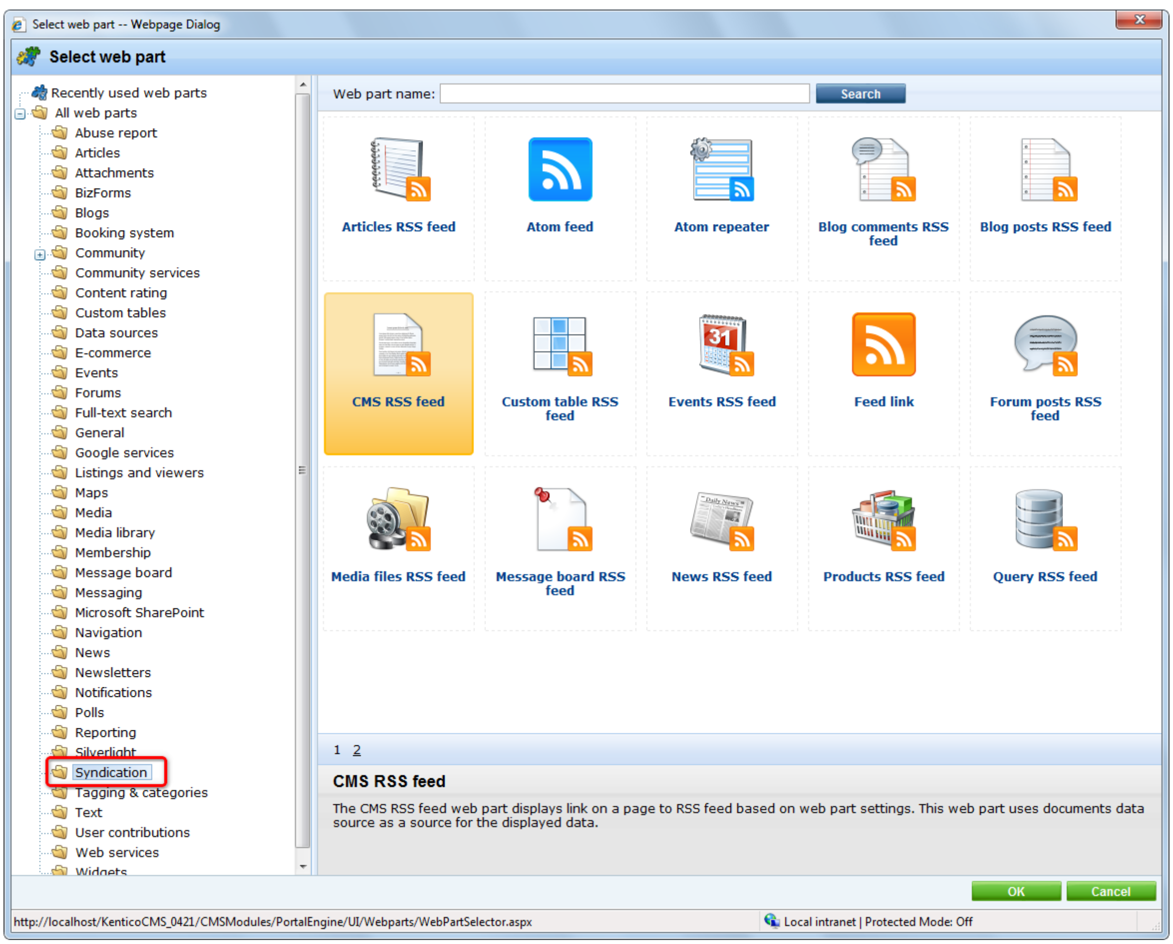
Apart from this universal web part, there are other web parts in the **Syndication** category which are pre-configured for one particular document type. See the listing in the Syndication web parts and widgets chapter.

In the following example, you will learn how to use this web part to create a feed link on the **/Products** page of the sample **Corporate site**. We will configure it so that it provides an RSS feed with products stored in all three categories under the page. The same procedure can be used on any other page and for documents of any other type, you just need to specify your required path, document type and transformation in the web part properties.

1. Sign in to CMS Desk and select the **/Products** page from the content tree. Switch to the **Design** tab and click the **Add web part** () icon of the *zoneLeft* web part zone.



2. In the **Select web part** dialog, select the **Syndication** category. Choose the **CMS RSS feed** web part and click **OK**.



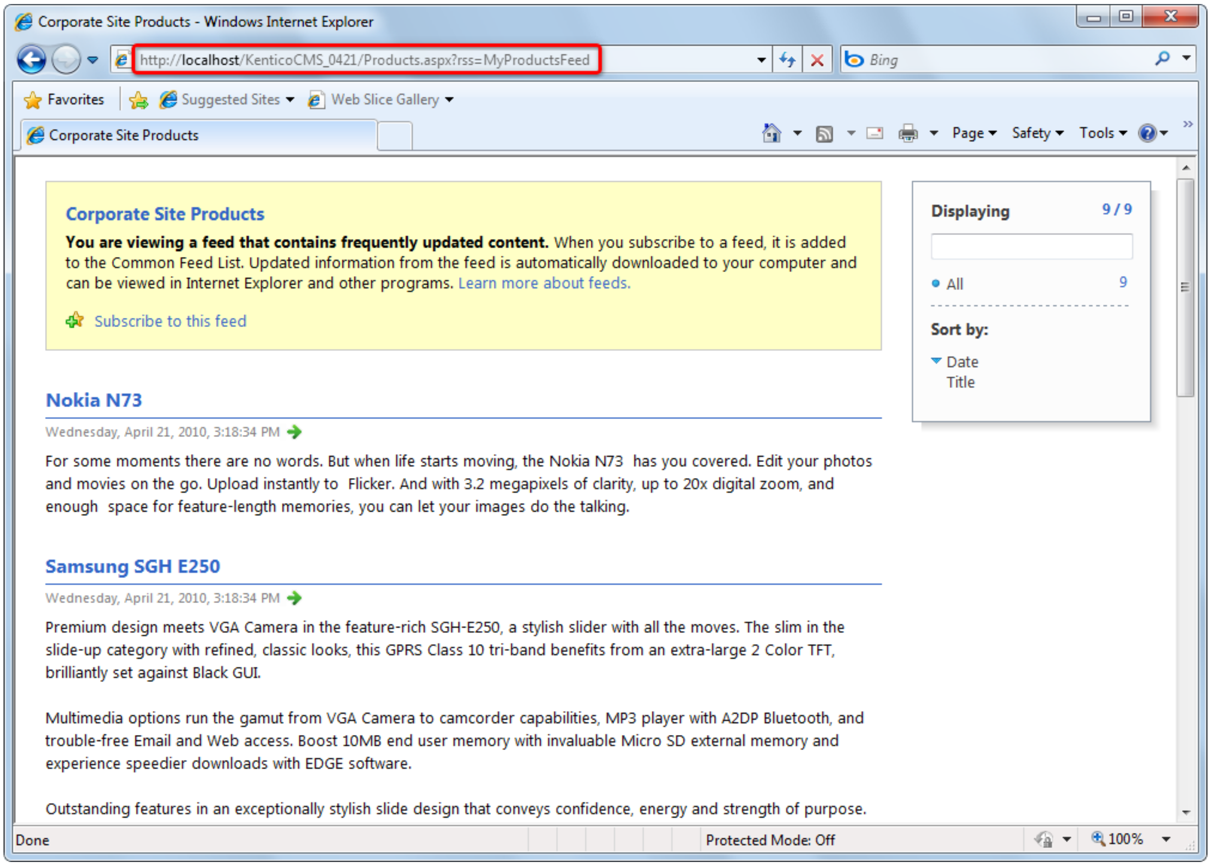
3. The **Web part properties** dialog opens. To achieve the required behavior, use the following configuration:

- **Link text**: *Products RSS feed*
- **Feed name**: *MyProductsFeed*
- **Feed title**: *Corporate Site Products*
- **Feed description**: *This is a sample RSS feed of products on the Corporate Site.*
- **Path**: */Products/%/*
- **Transformation name**: *CMS.RSSTransformations.Products*
- **Document types**: *CMS.CellPhone;CMS.Laptop;CMS.Pda*

Leave default values for the rest of the properties and click **OK**.

4. Sign out of CMS Desk and browse to the **Products** page. You should see the RSS icon and link as highlighted in the screenshot below.

5. Click the icon or link, your browser will detect that you are accessing an RSS feed and display its content. As you can see, the URL of the feed is actually the URL of the page where the web part is placed with the **?rss=MyProductsFeed** querystring appended (this can be modified using the *Feed querystring key* and *Feed name* web part properties). The same URL can be used to access the feed directly both from a browser or a dedicated RSS reader program.



8.34.4.2 RSS feed + data source

By connecting the **RSS feed** web part to a **data source**, you can create an RSS feed of the data provided by the data source. The **Atom feed** web part can be used exactly the same way, but the rendered feed is in the *Atom* format.

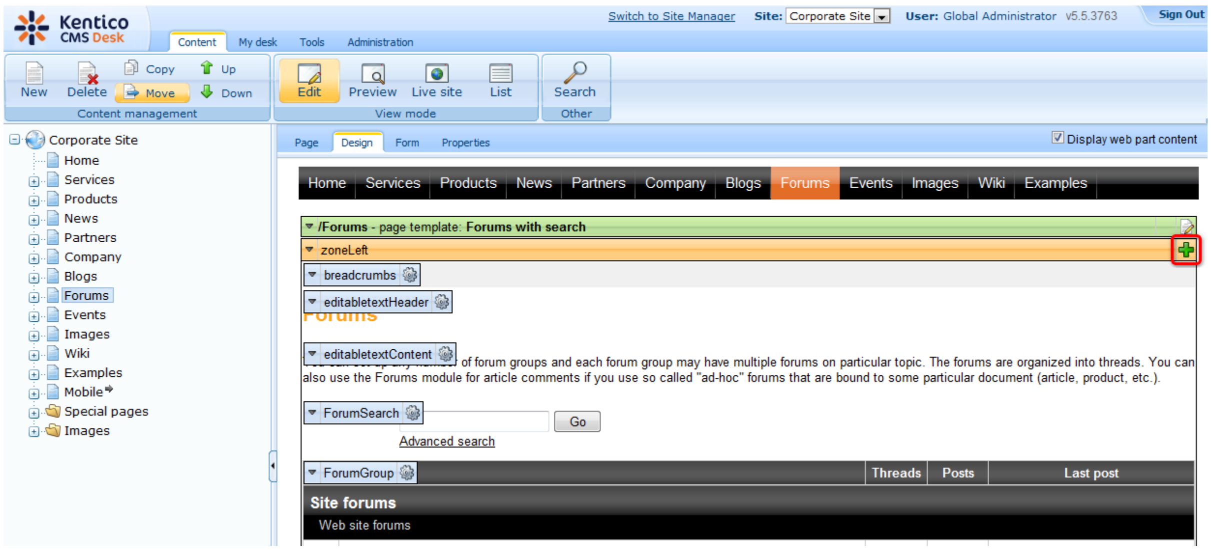
In the previous example, we have demonstrated how to use the *CMS RSS Feed* web part to create an RSS feed of **documents** stored in the content tree. The approach that we will explain now can also be used for that purpose, but you will typically use it to create feeds from Kentico CMS **objects** (forum posts, board comments, media files, ...).

In the following example, you will see how to create an RSS feed of forum posts in the site forums on the sample Corporate site. We will achieve this using the **RSS feed** and **Forum posts data source** web parts. The same procedure can be used for any other data, you only need to use a properly configured data source and handle the provided data by a suitable transformation.

> **Please note**
>
> The result of the following example can also be achievable using a single web part - the **Forum posts RSS feed** web part, which has a built-in *Forum posts data source*. Similarly, there are all-in-one web parts for other frequently used document types and objects (see the listing here).

> The following example uses two separate web parts for educational purposes only.

1. Sign in to CMS Desk and select the **/Forums** page from the content tree. Switch to the **Design** tab and click the **Add web part** () icon of the *zoneLeft* web part zone.



2. Select the **Data sources/Forum posts data source** web part and click **OK**. Leave all its properties at the default values, which will result in the data source providing all forum posts from all forum groups on the site. Click **OK**. The data source will be added to the bottom of *zoneLeft*.

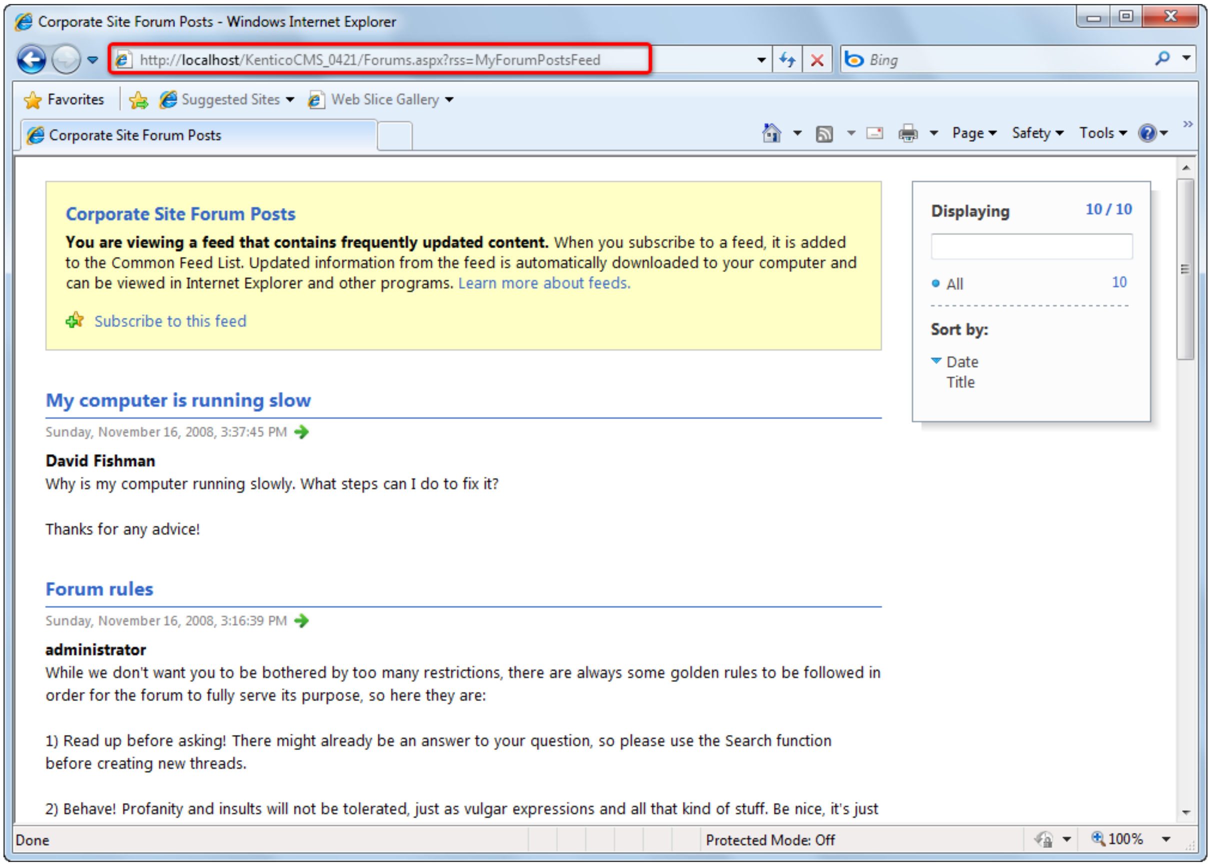
3. Click the **Add web part** () icon again and select the **Syndication/RSS feed** web part. Click **OK** and enter the following in the *Web part properties* dialog:

- **Link text**: *Forum posts RSS feed*
- **Feed name**: *MyForumPostsFeed*
- **Feed title**: *Corporate Site Forum Posts*
- **Feed description**: *This is a sample feed of all forum posts on the Corporate Site.*
- **Data source name**: *ForumPostsDataSource*
- **Transformation name**: *CMS.RSSTransformations.ForumPosts*

Leave defaults for the rest of the properties and click **OK**. The two web parts should now be at the bottom of the web part zone and fully configured for the required behavior.

4. Sign out of CMS Desk and browse to the **Forums** page. You should see the RSS icon and link as highlighted in the screenshot below.

5. Click the icon or link, your browser will detect that you are accessing an RSS feed and display its content. As you can see, the URL of the feed is actually the URL of the page where the web part is placed with the **?rss=MyForumPostsFeed** querystring appended (this can be modified using the *Feed querystring key* and *Feed name* web part properties). The same URL can be used to access the feed directly both from a browser or a dedicated RSS reader program.



8.34.4.3 RSS repeater + Data source

The **RSS repeater** web part can be used to transform a page into an RSS feed. This means that when a page containing the web part is accessed, its response type is changed to *application/xml* and the content of the feed is rendered. This is useful typically if you want your feed to have a dedicated URL (i.e. not a URL of a page with a querystring parameter). The **Atom repeater** and **XML repeater** web parts can be used exactly the same way, but the rendered feed is in the *Atom* or *XML* format.

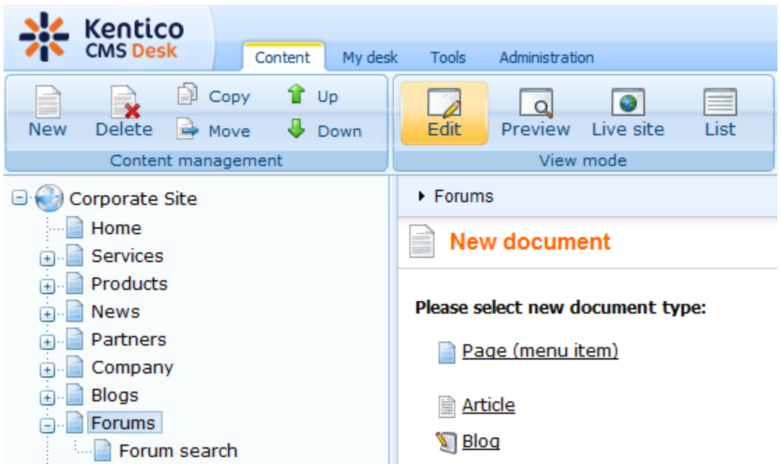
In the following example, we will create an RSS feed with forum posts, the same as in the previous example. But this time, we want the feed URL to be *<application path>/Forums/RSS.aspx*. Therefore, we will create a **dedicated page** with this URL which will contain the **RSS repeater** web part and which will be linked from the **/Forums** page.

Please note

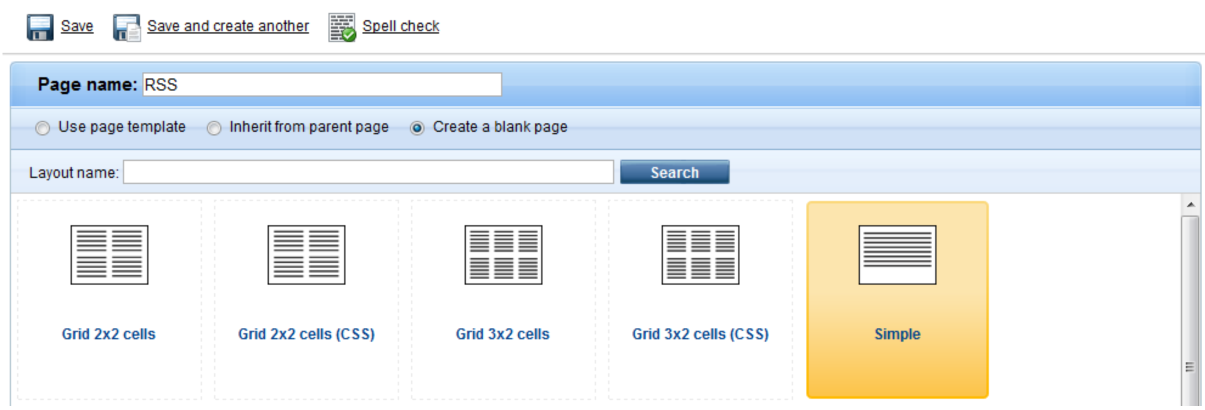
The result of the following example can also be achieved using a single web part - the **Forum posts RSS feed** web part, which has a built-in *Forum posts data source*. Similarly, there are all-in-one web parts for other frequently used document types and objects (see the listing here).

The following example uses two separate web parts for educational purposes only.

1. Sign in to CMS Desk and select the **Forums** page from the content tree. Click **New** and choose the **Page (menu item)** document type.



2. Choose to **Create a blank page** using the **Simple** layout. Enter *RSS* for **Page name** and click **Save**.

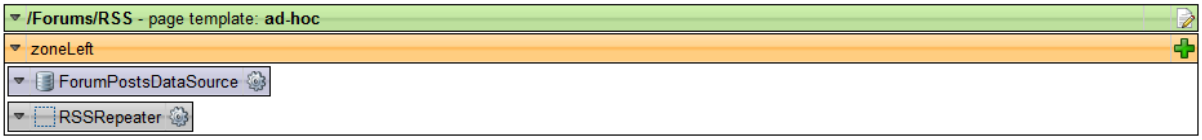


3. The page opens in **Design** mode. Click the **Add web part** (+) icon of the only web part zone on the page. Select the **Data sources/Forum posts data source** web part and click **OK**. Leave all its properties at the default values, which will result in the data source providing all forum posts from all forum groups on the site. Click **OK**.

4. Click the **Add web part** (+) icon again and select the **Syndication/RSS repeater** web part. Click **OK** and enter the following in the *Web part properties* dialog:

- **Feed name**: *MyForumPostsFeed*
- **Feed title**: *Corporate Site Forum Posts*
- **Feed description**: *This is a sample feed of all forum posts on the Corporate Site.*
- **Data source name**: *ForumPostsDataSource*
- **Transformation name**: *CMS.RSSTransformations.ForumPosts*

Leave defaults for the rest of the properties and click **OK**.



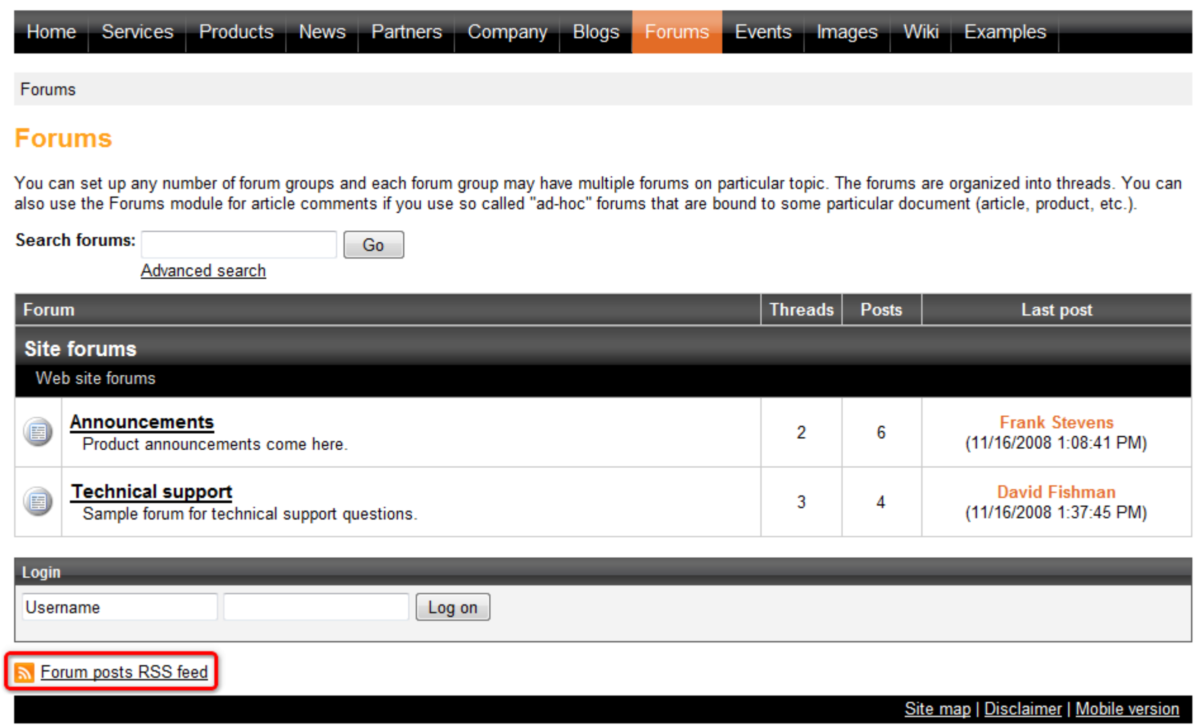
5. Now we have the RSS feed created, but without the knowledge of the page's URL, site visitors have no chance to access it. This is exactly when the **Feed link** web part comes in handy. It is a simple web part which displays the RSS icon with a link leading to a URL specified in its properties. We will add it to the */Forums* page.

Select the **/Forums** page in the content tree, switch to the **Design** mode and click the **Add web part** ( ) icon of the *zoneLeft* web part zone. Select **Syndication/Feed link** and click **OK**. Enter the following properties:

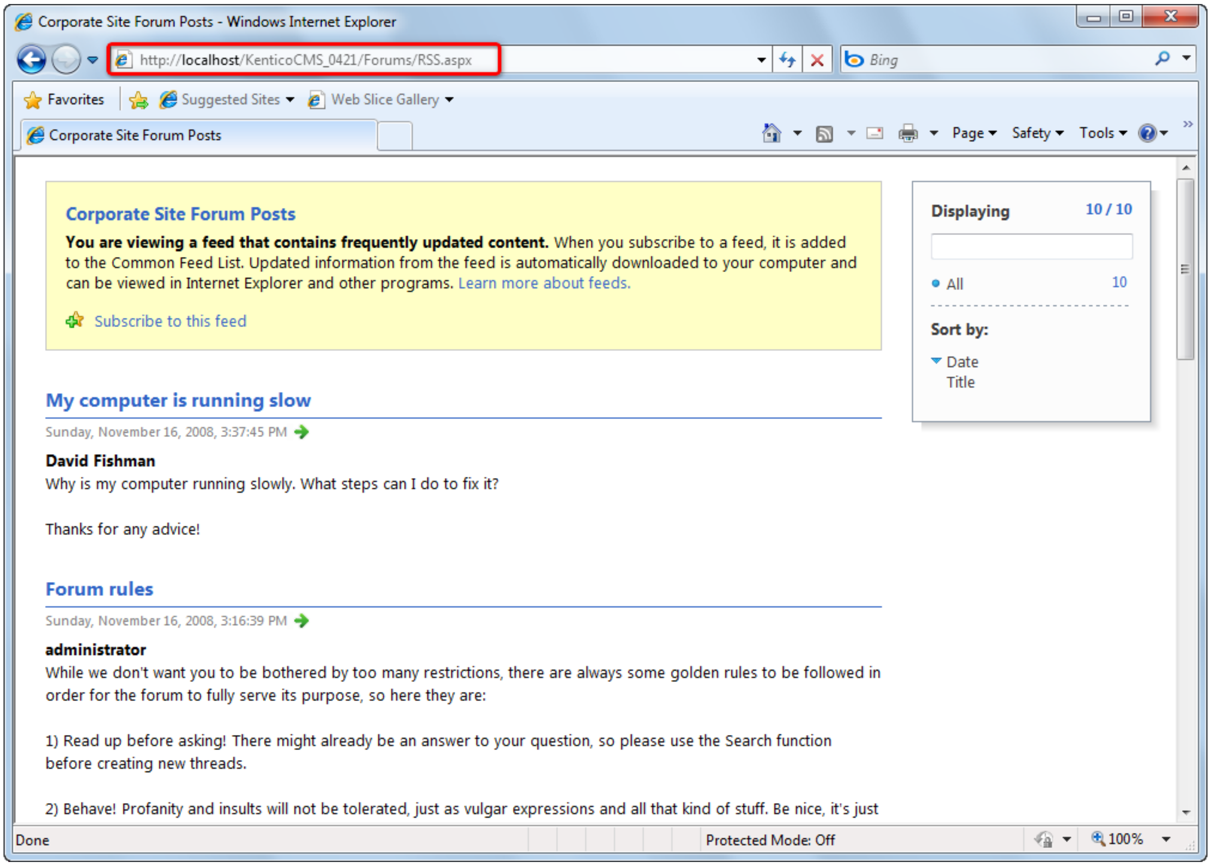
- **Link text**: *Forum posts RSS feed*
- **Feed URL**: *~/Forums/RSS.aspx*
- **Feed title**: *My Forum Posts Feed*

Leave defaults for the rest of the properties and click **OK**.

6. Sign out of CMS Desk and browse to the **Forums** page. You should see the RSS icon and link as highlighted in the screenshot below.



7. Click the icon or link, your browser will detect that you are accessing an RSS feed and display its content. As you can see, the URL of the feed is actually the URL of the page where the web part is placed. The same URL can be used to access the feed directly both from a browser or a dedicated RSS reader program.



8.34.5 Creating RSS feed pages manually (obsolete)

Due to backward compatibility with Kentico CMS versions prior to 5.5, it is also possible to create a dedicated page with an RSS feed. This way of creating RSS feeds is **obsolete** now, as syndication web parts provide a much more convenient way of creating RSS feeds.

The default installation contains a simple **CMSPages\NewsRss.aspx** page which shows how to build your own RSS feed. It works with news items, but you can modify the code so that it displays a different type of documents.

The following code example shows the code of the *NewsRss.aspx* page.

Please note: If you installed the Kentico CMS project as a web application, you need to rename the *CodeFile* property on the first line to *Codebehind* for the code example to be functional.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="RSS.aspx.cs"
    Inherits="RSSNews" %>
<%@ Register Assembly="CMS.Controls" Namespace="CMS.Controls"
    TagPrefix="cc1" %><?xml version="1.0" encoding="utf-8" ?>
<rss version="2.0">
 <channel>
  <title>News RSS</title>
    <link><![CDATA[<%=HttpContext.Current.Request.Url.AbsoluteUri.Remove
(HttpContext.Current.Request.Url.AbsoluteUri.Length - HttpContext.Current.Request.
```

```
Url.PathAndQuery.Length) + HttpContext.Current.Request.ApplicationPath%>]]></link>

  <description>News RSS Feed</description>
  <cc1:cmsrepeater ID="NewsRepeater" runat="server" OrderBy="NewsReleaseDate DESC"
 ClassNames="cms.news"
   TransformationName="cms.news.rssitem" SelectedItemTransformationName="cms.news.
rssitem"
   Path="/news/%"></cc1:cmsrepeater>
 </channel>
</rss>
```

As you can see, the page contains only RSS elements with dynamic code. The RSS items are rendered using a *CMSRepeater* control with appropriate transformation.

The code behind looks like this:

[C#]

```
protected void Page_Load(object sender, EventArgs e)
{
    Response.ContentType = "text/xml";
}
```

This code changes the output content type to XML.

How to create an RSS feed page for a different document type

If you want to display articles instead of news in your RSS feed, you will need to follow these steps:

1. Create a new ASPX page called articles_rss.aspx.

2. Copy and paste all code from the *NewsRss.aspx* file except for the <%@ Page %> directive.

3. Change the following properties of the CMSRepeater control:

- **OrderBy**: DocumentModifiedWhen DESC
- **ClassName**: cms.article
- **TransformationName**: cms.article.rssitem
- **SelectedItemTransformationName**: cms.article.rssitem
- **Path**: /%
- **WhereCondition**: leave empty

4. Add the same line of code as used in the *NewsRss.aspx.cs* code behind file (*Response.ContentType = "text/xml"*) to articles_rss.aspx.cs.

5. Create the trasformation **cms.article.rssitem** like this in **Site Manager -> Development -> Document Types -> ... edit Article ... -> Transformations**:

```
<item>
```

```
    <guid isPermaLink="true"><![CDATA[<%# GetAbsoluteUrl(GetDocumentUrl()) %>]]
></guid>
    <title><![CDATA[<%# Eval("ArticleTitle") %>]]></title>
    <description><![CDATA[<%# Eval("ArticleText") %>]]></description>
    <pubDate><%# Convert.ToDateTime(Eval("DocumentModifiedWhen")).ToString("r") %
></pubDate>
    <link><![CDATA[<%# GetAbsoluteUrl(GetDocumentUrl()) %>]]></link>
</item>
```

This code renders the particular items.

8.35 SharePoint integration

8.35.1 Overview

Kentico CMS allows you to access data stored on a SharePoint (Windows SharePoint Services 3.0 (WSS) or MOSS - Microsoft Office SharePoint Server) server and display them on your site. This can be done using the web parts of the Sharepoint module.

The Usage examples sub-chapter contains sample tutorials on using the SharePoint web parts for various tasks.

Default logon settings, used to access a SharePoint server, can be configured as described in the Settings topic.

What it can do

- Get list and list item data from SharePoint and provide it to Kentico CMS.
- Download documents or images.

What it can't do

- Modify SharePoint data.
- Display Kentico CMS data in SharePoint.

How it works

SharePoint web parts use pre-generated proxy classes of selected SharePoint web services to get data from the server. The following services are used:

- *http://server/_vti_bin/Lists.asmx* - methods for working with lists
- *http://server/_vti_bin/Imaging.asmx* - methods for working with picture libraries
- *http://server/_vti_bin/Copy.asmx* - methods for retrieving file content

The returned data is in CAML (XML) format, which needs to be further processed to display the data in a meaningful way. XSLT or ASCX transformations may be used for this purpose. For ASCX, the CAML response must be transformed to an ASP Dataset.

8.35.2 Web parts

Web parts for connecting to SharePoint are stored in the **Microsoft SharePoint** category in the web part catalog. This includes the following four web parts:

- **SharePoint datasource**
- **SharePoint datagrid**
- **SharePoint datalist**
- **SharePoint repeater**

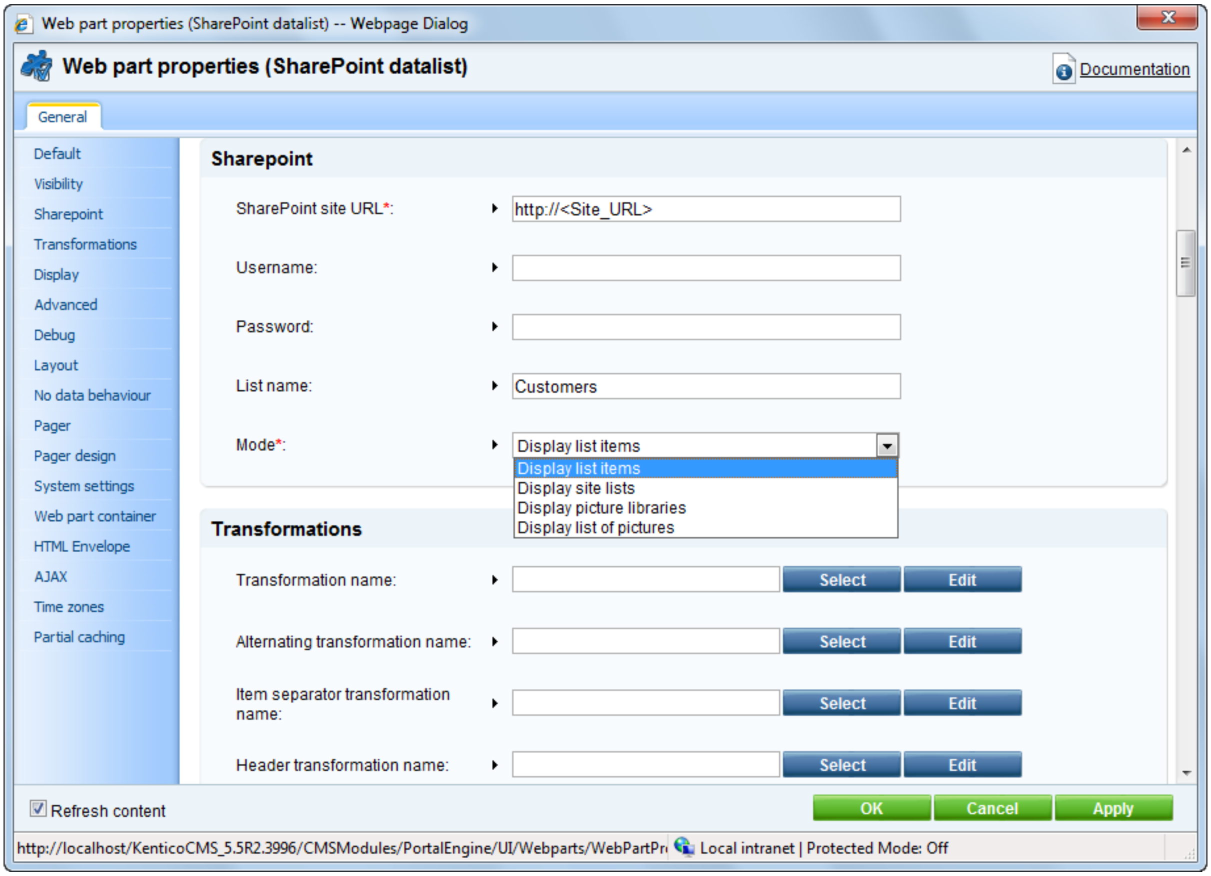
In this text, you can find information about the specific properties of these web parts.

SharePoint

The following properties can be configured in the **SharePoint** section. All the web parts listed above have these settings in common.

- **Mode**: determines what the web part will do (what data will be retrieved from the SharePoint server) and therefore which web service should be used
- **SharePoint site URL**: URL of the SharePoint site; based on the *Mode* settings, the full URL of the required web service is determined by this option (it is not configured directly)
- **Username, password**: you can specify the username and password for authentication directly in the web part properties, or you can inherit global values from **Site Manager -> Settings -> Microsoft SharePoint**; if you want to use Site Manager settings, leave the fields empty
- **List name**: if you use the *Display list items* or *Display list of pictures options*, you must specify the list from which the items should be loaded; in the other two modes, you can leave it empty; please note: the field is case sensitive

Just these settings should make the **SharePoint datagrid** display data or the **SharePoint datasource** to provide them. The other two web parts still need to have a transformation specified.



Transformations

The **SharePoint datalist** and **SharePoint repeater** need to have a transformation specified in order to display the retrieved data.

There is a number of pre-defined transformations which you can use almost right-off and of course modify them according to your needs. The transformations are stored under the **SharePoint - Transformations** document type (can be edited in **Site Manager -> Development -> Document types**). Their names are self-explaining - they correspond to what they should be used for.

The only thing you need to do to get them to work is edit ( ) them and **replace the sample server or field names with the real names** used on the SharePoint server.

Advanced

In the **Advanced** section, you can specify (limit) in more detail what you want to be retrieved. These settings are applied only in the “Get list items” mode.

- **Row limit**: sets the maximum number of retrieved items
- **Query**: filters or sorts returned items; it is similar to WHERE conditions in SQL, but it has totally different syntax based on CAML; for more information on how to construct such queries, search SharePoint documentation or use Google with keywords like “sharepoint query”.
- **View fields**: specifies directly which fields (columns) of a SharePoint list should be retrieved

Display

In this section, you can control how the retrieved data will be processed and displayed.

- **Selected item querystring key name**: name of the querystring key by whose presence the items will be selected
- **Selected item field name**: allows you to specify by which field you want the items to be selected; the default approach is to select items by ID - the web part tries to get query parameters with the name specified in the *Selected item querystring key* property and uses its value, e.g. *.aspx?id=2* will select the item with the value of its ID field equal to 2 if you entered *id* into the property above; note: the original field name is used here – without the *ows_* prefix; the field is case sensitive
- **Selected item field type**: provides information about the type of the *ItemID* field; for IDs, you should use the *Counter* type; for strings, use the *Text* type; these two should be sufficient for most cases, but you can search SharePoint documentation for other types
- **Use dataset**: switches between the usage of ASCX and XSLT transformations; if enabled, retrieved XML is converted to a dataset which serves as a data source for a classic ASP repeater; using XSLT, you can convert XML to HTML directly
- **Dataset fields**: if you want to use the dataset, but you don't need all fields, you can enumerate the fields which should be included in the dataset, separated by semicolons; this property can be particularly useful if you want to use cache

<u>Please note</u>: Properties in the **Display** section are not available for the **SharePoint datagrid** web part.

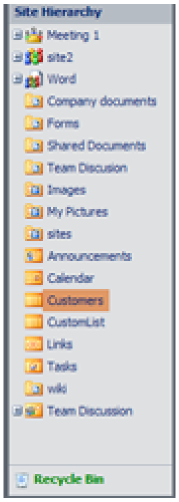
Debug

Because it is sometimes hard to tell or remember how SharePoint addresses list fields, we implemented the **Show raw response** function. It displays all retrieved SharePoint data in its raw XML form. You can look for fields which contain valuable information and can be displayed. Note: if the data is fetched from cache, the response can't be printed.

8.35.3 Usage examples

8.35.3.1 Site hierarchy

In this example, you will see how to display site lists which resemble the site hierarchy panel in the SharePoint site. It is quite simple. In fact, it only requires setting the *Display site lists* mode.



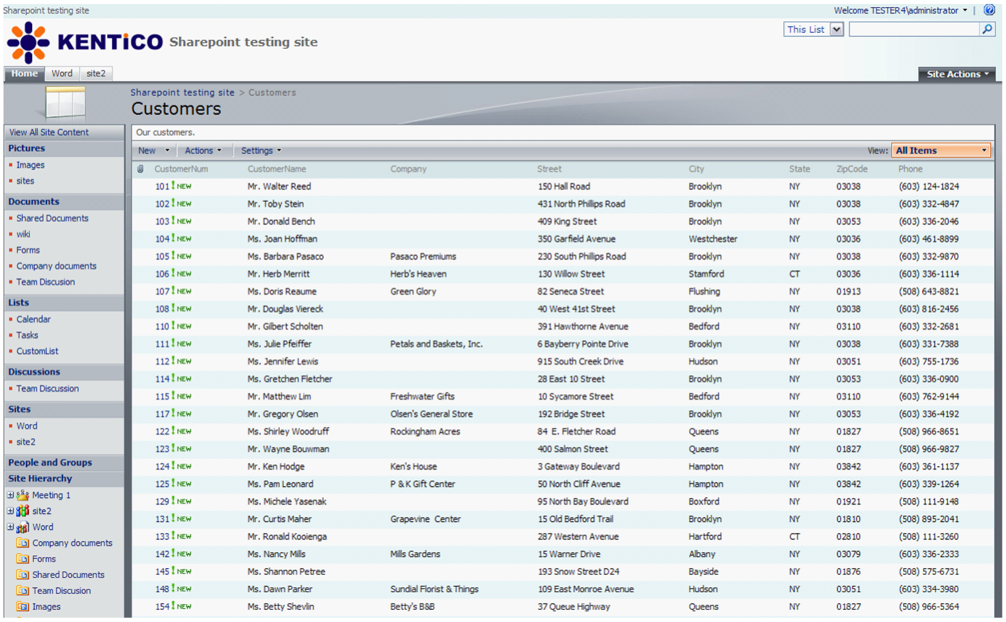
This can be done by adding one of the SharePoint web parts (e.g. **Microsoft SharePoint -> SharePoint repeater**) and configuring them the following way:

- **SharePoint site URL**: URL of your SharePoint server
- **Mode**: Display site lists
- **Username, password**: enter your username and password or leave the fields empty if you have them configured in **Site manager -> Settings -> Microsoft SharePoint**
- **Transformation**: there is a prepared transformation for displaying a site hierarchy called **SiteHierarchy** under the **SharePoint – Transformations** document type. For correct behavior, you must only change the name of the SharePoint server.

Leave default values for the rest of the properties and click **OK**. You should see a result similar to the screenshot below - it simply creates a link to each of the lists on the SharePoint server.

8.35.3.2 List items

The *Display list items* mode enables you to display items from any list from the SharePoint server.

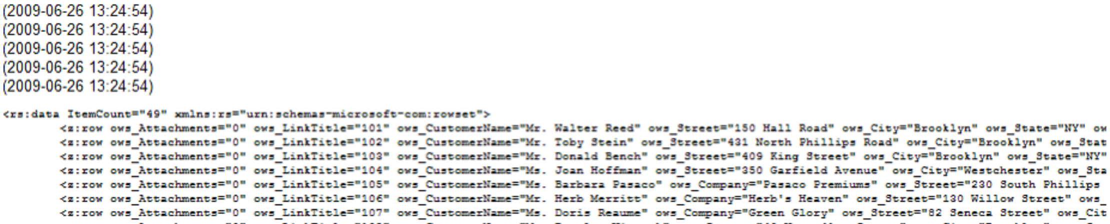


Let's say that in SharePoint, there is a custom list called *Customers*, storing information about company customers in fields like *Customer name*, *Street*, *City*, *Phone*, etc. Let's say that we want to display data from this list on our Kentico CMS website.

1. This can be done by adding one of the SharePoint web parts (e.g. **Microsoft SharePoint -> SharePoint repeater**) and configuring them the following way:

- **SharePoint site URL**: URL of your SharePoint server
- **Mode**: Display list items
- **Username, password**: enter your username and password or leave the fields empty if you have it configured in **Site manager -> Settings -> Microsoft SharePoint**
- **Transformation**: there is a prepared transformation for item lists called **ListItems** under the **SharePoint – Transformations** document type; for correct behavior, you must only change the name of the SharePoint server
- **Show raw response**: enabled; this is because we will need to inspect the retrieved data and change the transformation to suit our needs

Leave default values for the rest of the properties and click **OK**. You should see a result similar to the screenshot below.



2. That's not exactly what we want. We would like to see the customers' name and city. We must look into the raw output and search for those fields. We find them there under the *ows_CustomerName* and *ows_Street* attributes.

Now that we know the names of the attributes, we need to edit the transformation code or create a new transformation.

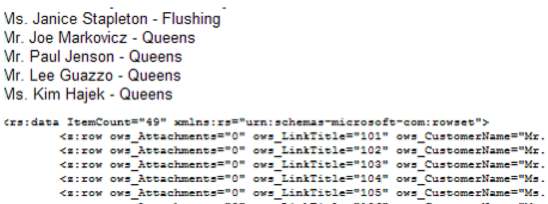
Replace the original **Transformation**:

```
<%# Eval("ows_Title") %> (<%#  Eval("ows_Created")%>) <br />
```

With something like this:

```
<%# Eval("ows_CustomerName") %> - <%#  Eval("ows_City")%> <br />
```

The output looks better now, it displays the desired values.



3. Now we would like to be able to click each item and see more detail about it. We must prepare the **Selected item transformation**. The transformation may look like the following code sample:

```
<strong><%# Eval("ows_CustomerName") %></strong><br/>
<%# Eval("ows_City")%> <br/>
<%# Eval("ows_Street")%> <br/>
<%# Eval("ows_Phone")%> <br/>
```

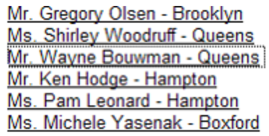
4. In web part properties, we must enter the field name of the field by which we want to select items. In this case, it is the *ID* field. In raw response output, it can be seen as *ows_ID*. So enter the following values:

- **Selected item querystring key name**: id
- **Selected item field name**: ID
- **Selected item field type**: Counter

Finally we must alter the original **Transformation** to suit the values we just entered, i.e. to create a link to customer details using the *id* parameter in query string. It can look like this.

```
<a href="<%# CMS.GlobalHelper.UrlHelper.AddParameterToUrl(CMSContext.RawUrl,"id",
(string)Eval("ows_ID")) %>">
<%# Eval("ows_CustomerName") %> - <%# Eval("ows_City")%> </a><br />
```

The final result looks like this - a simple text list of customers with links:



After clicking a customer name, a detail of the customer is displayed.

5. Now let's configure the **Advanced** settings. We will want a maximum of 10 customers to be displayed and we want to display only customers from *Queens*.

Enter number 10 into the **Row limit** field and use this code as the **Query**:

```
<Query>
<Where><Eq><FieldRef Name="City" /><Value Type="Text">Queens</Value></Eq></Where>
</Query>
```

The <*Eq*> tag means equals; field name is *City*, and its value of type *Text* should be equal to *Queens*.

Click **OK**, you should see the filtered output now as in the following screenshot:

8.35.3.3 Picture libraries

In this example, we use SharePoint to retrieve only a picture library type list. For setup, you must only set the following properties:

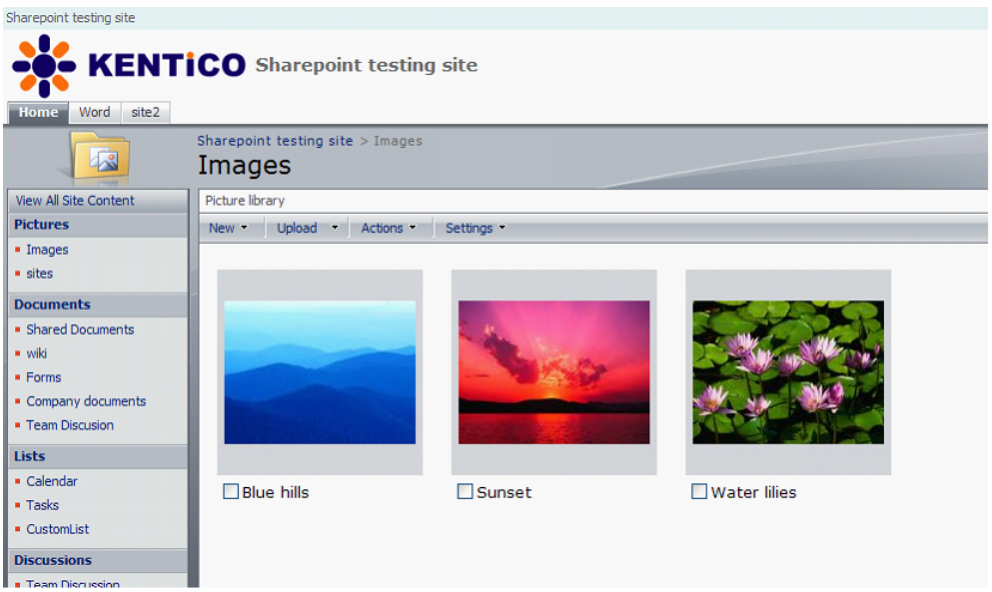
- **SharePoint site URL**: URL of your SharePoint server
- **Mode**: Display picture libraries
- **Username, password**: enter your username and password or leave the fields empty if you have it configured in **Site manager -> Settings -> Microsoft SharePoint**
- **Transformation**: there is a prepared transformation for picture libraries called **PictureLibLists** under the **SharePoint – Transformations** document type. For correct behavior, you must only change the name of the SharePoint server.

Leave default values for the rest of the properties and click **OK**. You should see a result similar to the screenshot below - it shows a list of picture libraries with links to the SharePoint server.

Empty
Images
My Pictures
sites

8.35.3.4 List of pictures

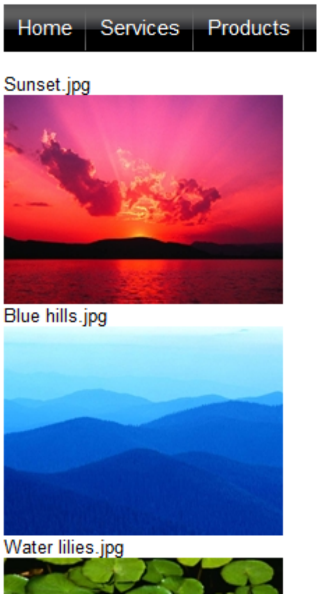
The *Display list of pictures* mode enables you to display pictures from picture libraries on a SharePoint server.



On the SharePoint server, we have a picture library named *Images* and 3 pictures in it. We want to display these pictures in our Kentico CMS site. For setup, you must enter the following properties of the web part (e.g. SharePoint repeater):

- **SharePoint site URL**: URL of your SharePoint server
- **Mode**: Display list of pictures
- **Username, password**: enter your username and password or leave the fields empty if you have it configured in **Site manager -> Settings -> Microsoft SharePoint**
- **List name**: *Images*
- **Transformation**: there is a prepared transformation for picture lists called **Pictures** under the **SharePoint – Transformations** document type. For correct behavior, you must only change the name of the SharePoint server.

Leave default values for the rest of the properties and click **OK**. You should see a result similar to the screenshot below.



The transformation code looks like this:

```
<%# SharePointFunctions.SplitSharePointField((string)Eval("ows_FileLeafRef"),1) %>
<br/>
<img src="<%# SharePointFunctions.GetSharePointFileUrl("tester4",
SharePointFunctions.SplitSharePointField((string)Eval("ows_FileRef"),1)) %>
&maxsidesize=200" /> <br />
```

The images are downloaded through the **GetSharePointFile.aspx** page, same as documents. So we create <*img*> tags with *src* attributes pointing to this page in the transformation. For ease of use, there is the **GetSharePointFileUrl** function in the **SharePointFunctions** class, which takes two parameters and returns the URL which downloads the file. The first parameter is server name (or site url), the second parameter is the location of the file in SharePoint. The final URL looks like this:

- *~/ CMSModules/Sharepoint/CMSPages/GetSharePointFile.aspx?server=server_name&name=Images/Sunset.jpg*

Please note: Credentials from settings are always used for downloading the file when the **GetSharePointFile** page is used.

There is another useful function in the **SharePointFunctions** class called **SplitSharePointField**. SharePoint for some reason creates combined attribute values for some fields. It looks like this - *ows_FileLeafRef="2;#Blue hills.jpg"*. We would often like to use only a part of such a value. The **SplitSharePointField** function serves this purpose. The first argument is a combined value and the second one is the index of the part we want in the output.

8.35.3.5 GetSharePointFile page

The **GetSharePointFile.aspx** page is a special page located under the **~/CMSModules/Sharepoint/ CMSPages** directory. The correct URL to access the page looks like the following:

- *~/ CMSModules/Sharepoint/CMSPages/GetSharePointFile.aspx?*

server=server_name&name=Images/Sunset.jpg

It downloads the specified file from the SharePoint server and sends it further to the user. Credentials configured in **Site Manager -> Settings -> Microsoft SharePoint** are used for authentication by the web service. You can control the Content-Disposition HTTP header using the *disposition* query parameter.

For images, it supports *maxsidesize*, *width* and *height* parameters to control the image size. Kentico CMS cache is used for images, you can configure it in **Site Manager -> Settings -> website -> Cache images (minutes)**.

Important!

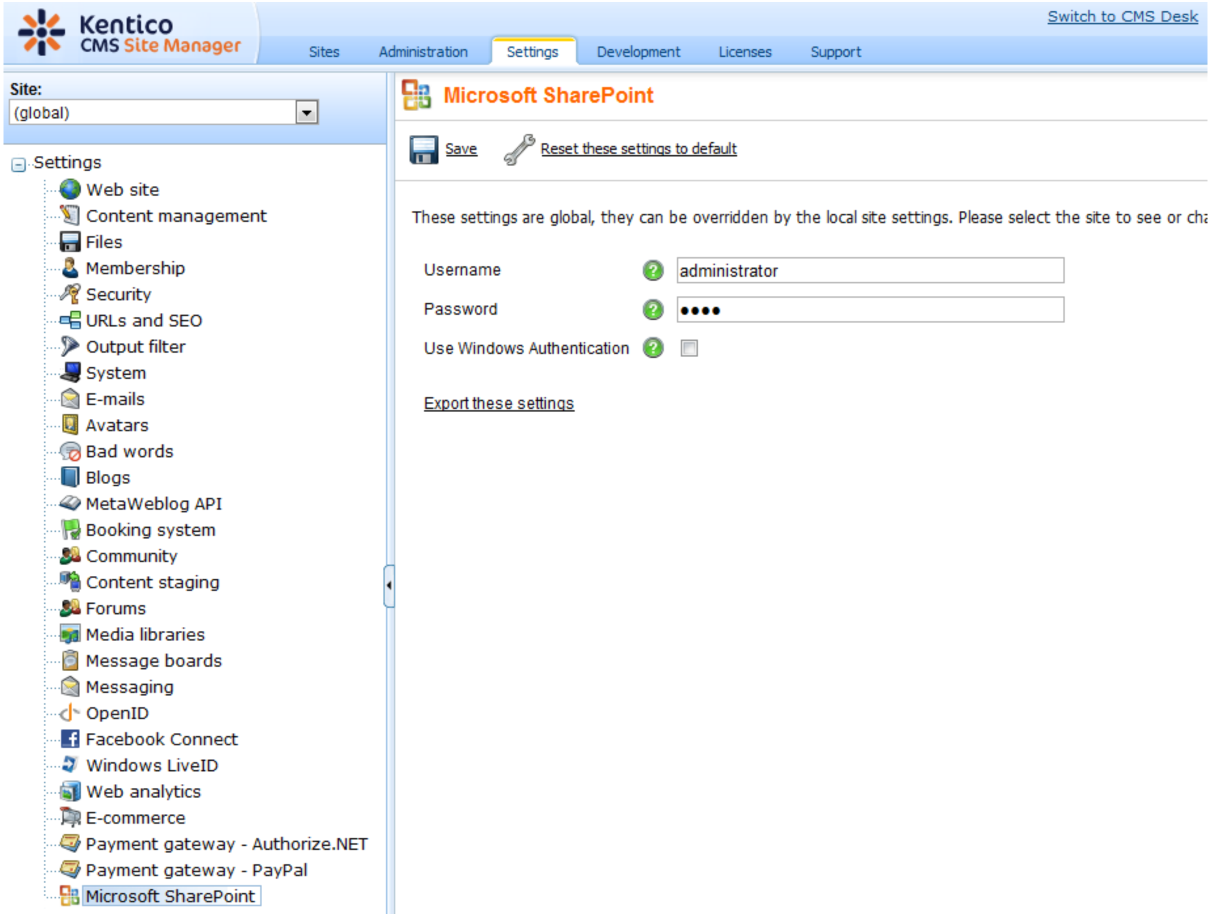
Because the settings configured in **Site Manager -> Settings -> Microsoft SharePoint** are automatically used for authentication by the **GetSharePointFile.aspx** page and since the URL to access the page can be entered manually, it is highly recommended to enter the credentials of a user that is authorized to access only the files you want to display on your website.

8.35.4 Settings

Default logon credentials used to access the SharePoint server can be configured in **Site Manager -> Settings -> Microsoft SharePoint**. If you configure a **Username** and **Password** here, it will be used by default if you leave the *Username* and *Password* fields **blank** in the properties of SharePoint web parts.

If you enable the **Use Windows Authentication** option, the current user's windows domain credentials will be used to access the SharePoint server.

The address of the SharePoint server itself needs to be specified in the properties of SharePoint web parts.



Important!

The settings configured here are automatically used for authentication by the GetSharePointFile.aspx page and since the URL to access the page can be entered manually, it is highly recommended to enter the credentials of a user that is authorized to access only the files you want to display on your website.

8.36 Smart search

8.36.1 Overview

The Smart search module enables index-based search of the content of your website. It is based on **Lucene.Net**, version 2.1.0, which is a source code, class-per-class, API-per-API and algorithmatic port of the Java **Lucene** search engine to the C# and .NET platform.

The module uses **indexes** to store information about the content of your site. When a search request is sent to the system by a user, it is the index that gets searched, which results in **significantly better performance** compared to linear SQL query search. For more information on **how the module works**, please refer to the How it works topic.

To enable the smart search functionality on your site, you need to perform the following three steps:

1. Enable Smart search indexing in the system. Learn how in the Enabling Smart search indexing topic.

2. Create an index. Assign it to a site (and culture) and define content that will be indexed. Learn how in the Managing indexes sub-chapter.

3. Use some of the Smart search web parts on your site.

The module comes with several **web parts**. You can find an overview of these web parts, including explanations of the most important web part properties, in the Available web parts and Using the Smart search filter topics.

The following topics provide additional information about search related issues:

- Search syntax
- Related scheduled tasks
- Searching attachments
- Security

In the past, Kentico CMS supported only linear SQL search functionality. To keep the system backward compatible, this functionality is still available. It is now referred to as **SQL Search** and you can find further information about it in the SQL Search overview topic.

The Smart search internals and API sub-chapter provides information about the database tables and classes used by the module and examples of how smart search indexes can be managed using the API and how search results can be displayed in transformations.

8.36.2 How it works

Information about the content specified for an index is stored in a physical **index file** on the local disk. The index files are located in **~/App_Data/CMSModules/SmartSearch/<Index code name>** within your web project's folder.

Documents, forums and other objects in Kentico CMS are reflected in the index file as **index documents**. The data structure of the index documents is much more suitable for being searched through, resulting in significantly higher search performance compared to linear SQL search.

The index documents may contain fields that the corresponding Kentico CMS objects contain, based on the settings made on each object type's **Search fields** tab. Depending on these settings, a representation of the object is created in the index file by the **Index writer**. When an object included in the index is created, removed or has one of its fields modified, the index document is updated. The **Index searcher**, on the other hand, searches through the index file and returns the relevant results from it.

Brief example

Let's take the following model scenario to explain the life cycle of a document in the index file:

1. A new document is created and stored in the database. When the fields to be indexed are defined for the document type, a new indexing task is logged in the database upon the document's creation.

2. The Smart search checks the database automatically on a regular basis for the presence of indexing tasks.

3. When the Smart search module finds the task, it processes it and the new document is added to the index.

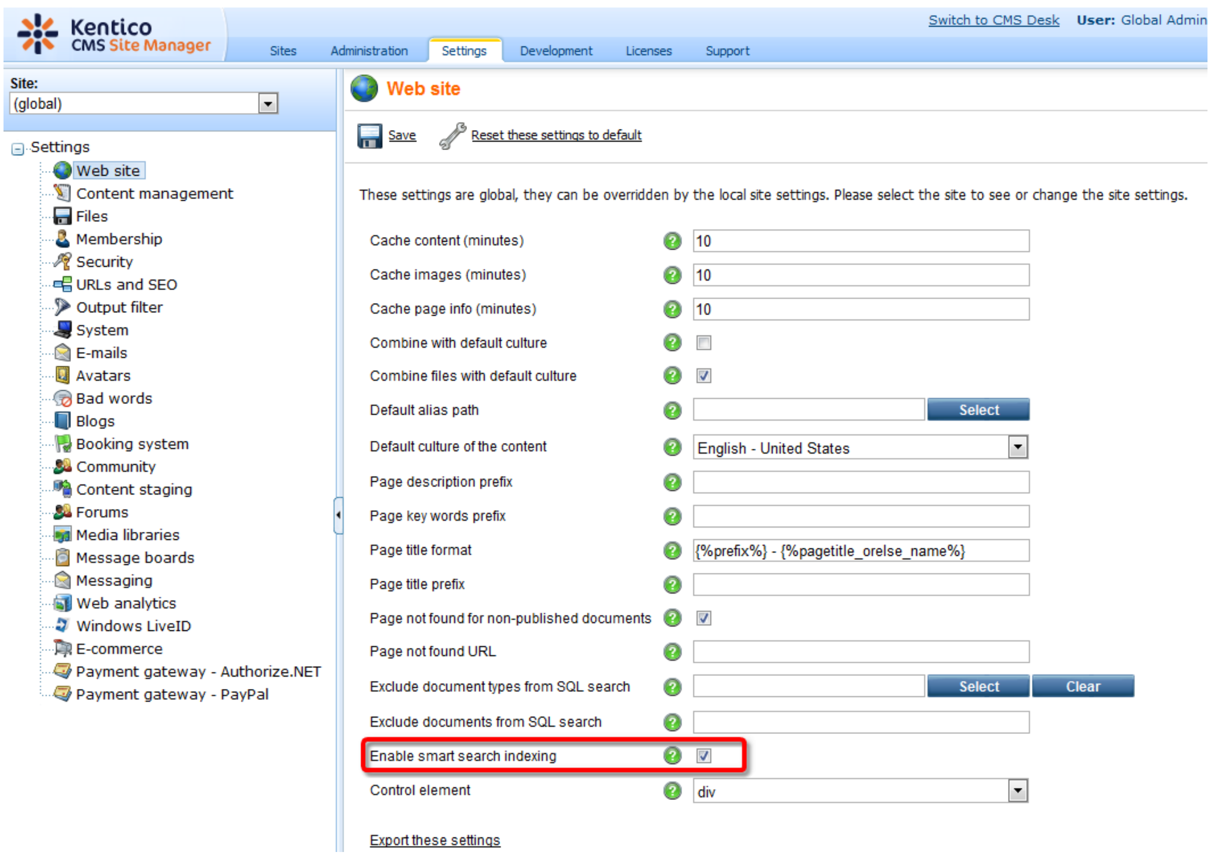
4. Now a user comes to the site and sends a search request via some Smart search web part.

5. The request gets passed to the Smart search module.

6. It modifies the request into a searchable form, searches the index and returns results based on what was found in the index.

8.36.3 Enabling Smart search indexing

To enable Smart search indexing, go to **Site Manager -> Settings -> Web site**, check the **Enable smart search indexing** check-box and click **Save**.



You also need to have the **write permission** assigned to the **~/AppData** folder. Indexes are stored within this folder, so the permission is necessary for the system to work with them. The procedure of granting a folder with the write permission is described in the Disk permission problems topic.

8.36.4 Managing indexes

8.36.4.1 Creating an index

Information about the content of your site is stored in indexes. You need to have the content of your site indexed in order for it to be searchable using the Smart search module.

There are four types of indexes, click the link to learn about how indexes of the type can be configured:

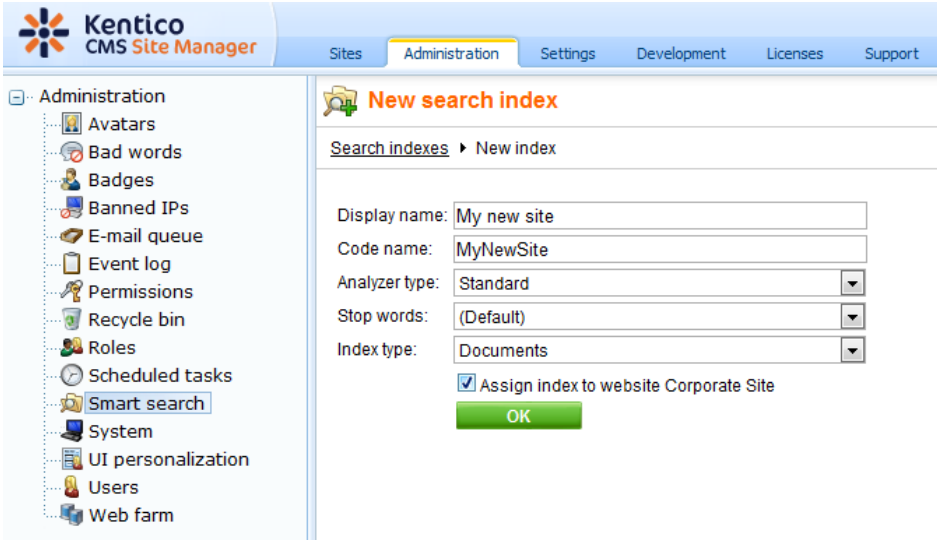
- Documents index - stores information about the content of documents in your content tree
- Forums index - stores information about the content of discussion forums on your site
- Custom table index - stores information about data stored in custom tables
- User index - stores information about website users

The following example shows the general procedure how an index can be created and what options you have when creating it. The procedure is applicable for both document and forum indexes, while differences between the two are noted in the text.

1. Go to **Site Manager -> Administration -> Smart search** and click the **New index** link. The **New search index** dialog will be displayed. You will be asked to enter the following details:

- **Display name**: name of the index displayed in the administration interface
- **Code name**: name of the index used in website code (the fully qualified file name must be less than 260 characters long, the directory name must be less than 241 characters long)
- **Analyzer type**: type of analyzer that will be used when indexing the content of your site, the following types are available:
  - *Standard* - grammar-based analyzer (stop-words, shortcuts, ...), very efficient for English, but may not produce satisfactory results with other languages
  - *Simple* - divides text at non-letter characters
  - *Stop* - contains a collection of stop-words at which it divides the text
  - *White space* - divides text at whitespace characters
  - *Keyword* - tokenizes the entire stream as a single token; this is useful for data like zip codes, ids, and some product names
- **Stop words** - dictionary containing words which will be omitted from indexing (e.g. 'and', 'or', ...) when a *Stop* or *Standard* analyzer is used; the dictionaries are stored in *~\App_Data\CMSModules\SmartSearch_StopWords*
- **Index type**: type of content to be indexed
  - *Documents* - indexes content of documents in the content tree
  - *Forums* - indexes content of discussion forums
  - *Users* - indexes details about system users - fields of the CMS_User system table
  - *Custom tables* - indexes records in custom tables
- **Assign index to site <sitename>** - if checked, the index will be assigned to the site whose name is displayed

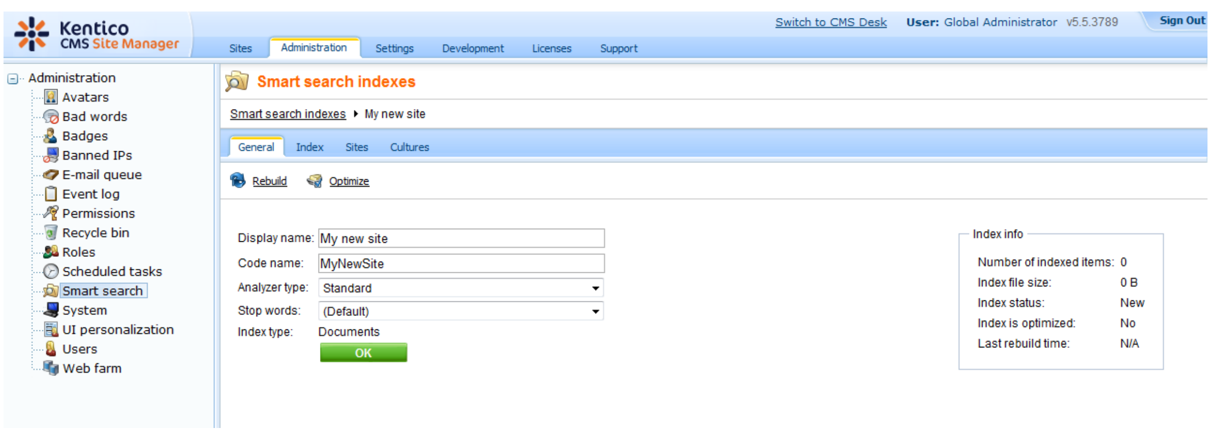
Click **OK**.



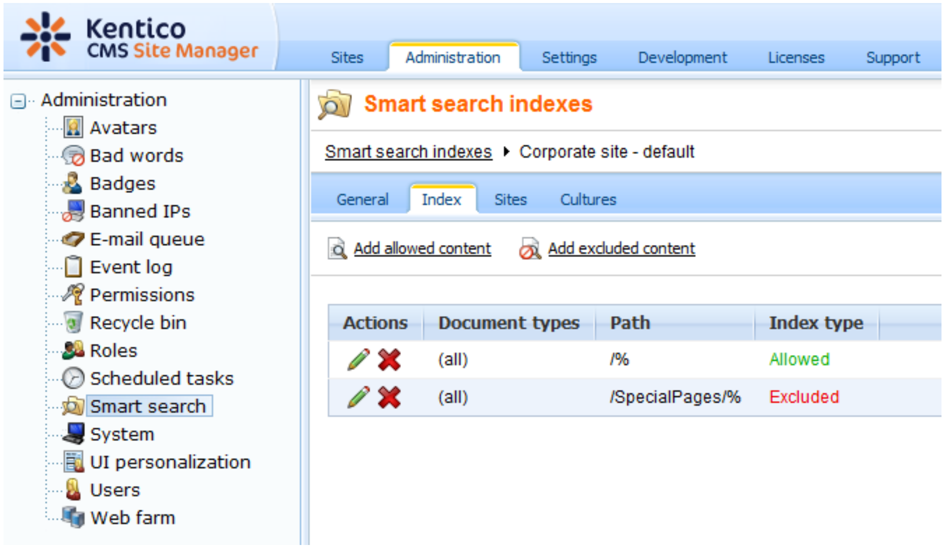
2. You will be redirected to the index's editing interface. The **General** tab, which you will be redirected to, shows the same details that you just entered. Notice the info box on the right, which shows general information about the index.

The following two actions can also be performed on this tab, but it would make no sense to perform them at this stage of creating a new index:

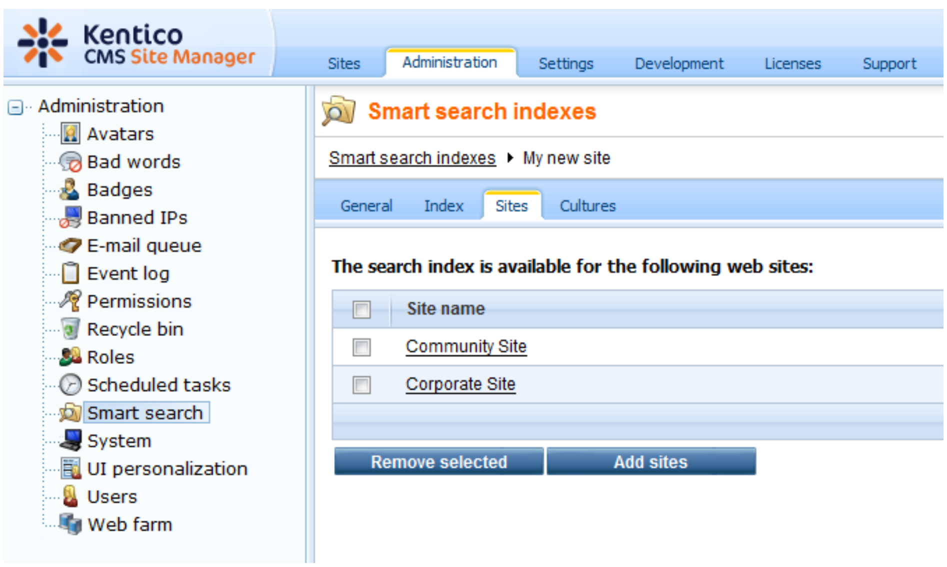
- **Rebuild** - deletes the original index and the specified content gets indexed again; clicking this action button does not always guarantee that the index will be rebuilt immediately, the process may be delayed if e.g. another index is already being rebuilt or if the rebuilding tasks are configured to be handled by the scheduler; the index is automatically optimized after a successful rebuild
- **Optimize** - de-fragments the index, which results in better search performance, particularly in the case of large indexes



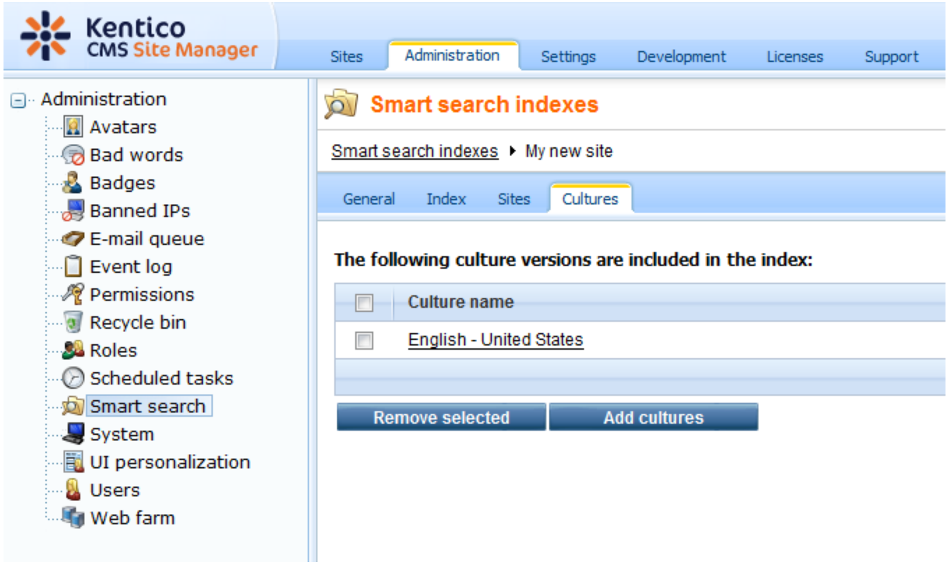
3. Switch to the **Index** tab. This is where you define what document content, forums, custom tables or users will be indexed. The content of this tab depends on the type of index that you are creating. Detailed information about defining index content is given in Defining document index content, Defining forums index content, Defining custom tables index content and Defining user index content.



4. Now switch to the **Sites** tab. Make sure that the index is assigned to the appropriate website. You may also want to optionally assign the index to some other site in order to have multi-site search results.



5. Switch to the **Cultures** tab (only available for document indexes). This is where you can choose which cultural versions of the site will be indexed by this index. Choose the cultural version(s) that you want to be indexed. If you have a multi-site index, you can select the cultures separately for each site chosen by the **Select site** drop-down list.



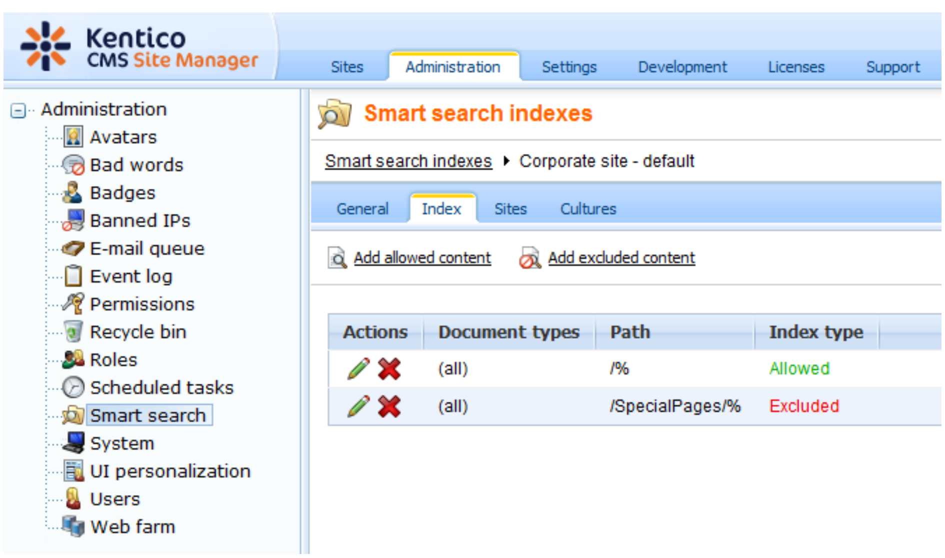
6. Finally, go back to the **General** tab and choose to **Rebuild** the index. This needs to be done only for the first time. Any further changes made to the site will be indexed automatically.

8.36.4.2 Defining document index content

On the **Index** tab of a **document index**, you can define which documents on the site will be indexed. This is done by defining allowed and excluded content.

Excluded content is subtracted from the allowed content. So for example if you allow **/%** and exclude **/Special-pages/%** at the same time, it means that all pages on the site will be indexed, except of all pages found under the **/Special-pages** node.

The dialogs for defining new allowed/excluded content can be accessed using the **Add allowed content** and **Add excluded content** links.



Adding allowed content

In this dialog, you are defining what part of the website will be indexed. You can specify this by choosing **Path** to the indexed documents, **Document types** of the indexed documents or a combination of both. The following options can be specified:

- **Path**: path to the documents that should be indexed
- **Document types**: document types that should be included in indexing

- **Including ad-hoc forums**: if checked, ad-hoc forums attached to documents specified by the properties above (if there are any) will be indexed too
- **Including blog comments**: if checked, blog comments attached to blog posts specified by the properties above (if there are any) will be indexed too
- **Including message boards**: if checked, message boards attached to documents specified by the properties above will be indexed too

Other objects or documents displayed on the specified documents by web parts will not be indexed. E.g. the content of news documents displayed by a Repeater web part on a specified document will not be added to the index etc.

Examples:

- **Path**: /%
- **Document types**: empty

In this case, all documents on the site will be indexed.

- **Path:** /Partners
- **Document types**: empty

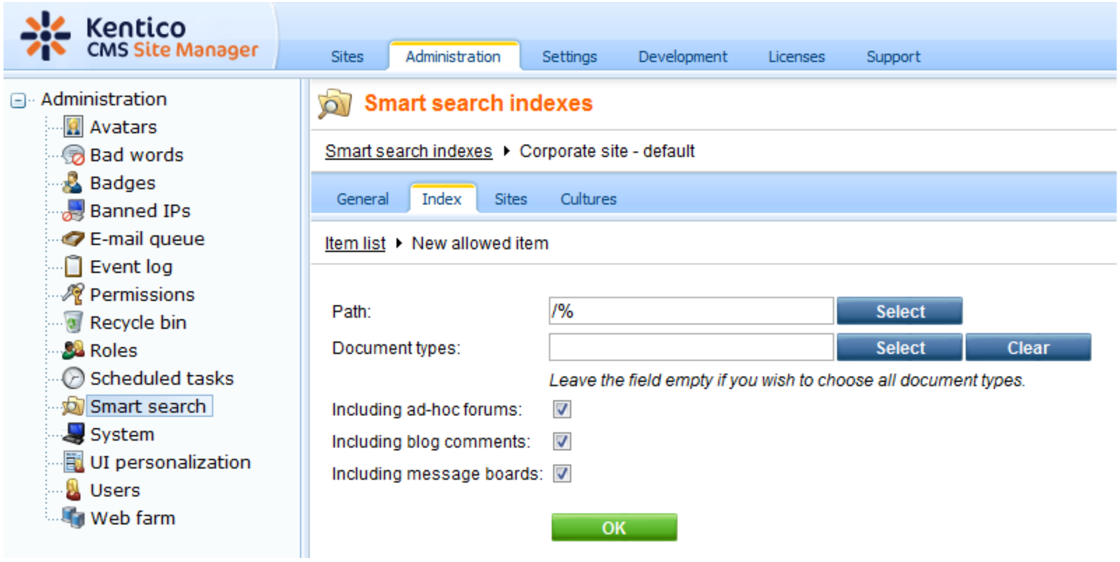
In this case, only the /Partners page will be indexed, without the pages under it.

- **Path**: empty
- **Document types**: CMS.News

In this case, all documents of the CMS.News document type on the whole site will be indexed. Please note that in this case, empty path field is equal to **/%**.

- **Path**: /Products/%
- **Document types**: CMS.CellPhone;CMS.Pda

In this case, all documents of the CMS.CellPhone and CMS.Pda document types found under /Products will be indexed.



New excluded content

In this dialog, you are defining what part of the website will be excluded from indexing.

Excluded content is excluded allowed content. So for example if you allowed **/%** and excluded **/Special-pages/%** at the same time, it means that all pages on the site will be indexed, except of all pages found under the **/Special-pages** node.

The following options can be specified:

- **Path**: path to the documents that should be excluded from the index
- **Document types**: document types that should be excluded from the index

Examples:

- **Path:** /Partners
- **Document types**: empty

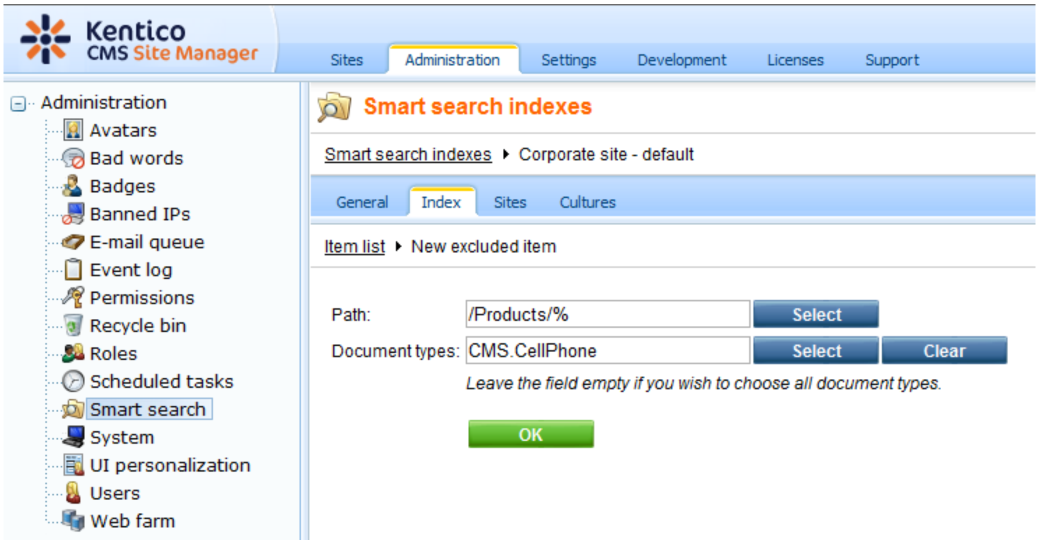
In this case, only the /Partners page will be excluded from indexing, not the pages under it.

- **Path**: empty
- **Document types**: CMS.News

In this case, all documents of the CMS.News document type on the whole site will be excluded from indexing. Please note that in this case, empty path field is equal to **/%**.

- **Path**: /Products/%
- **Document types**: CMS.CellPhone;CMS.Pda

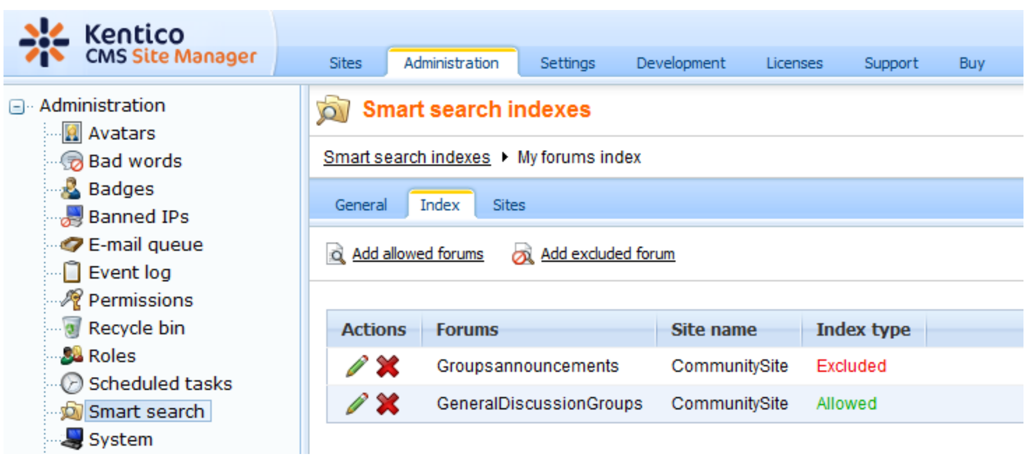
In this case, all documents of the CMS.CellPhone and CMS.Pda document types found under /Products will be excluded from indexing.



8.36.4.3 Defining forums index content

On the **Index** tab of a **forum index**, you can define which documents on the site will be indexed. This is done by defining allowed and excluded forums.

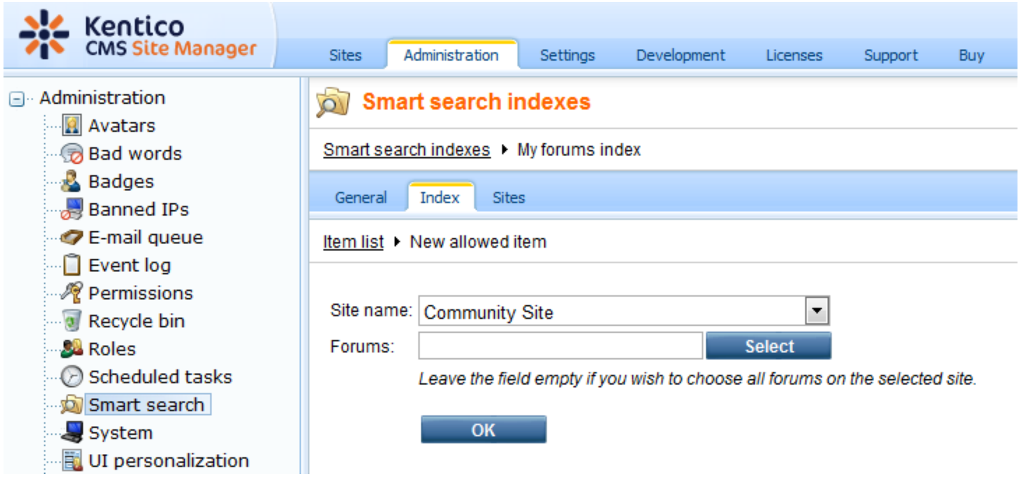
The dialogs for defining new allowed/excluded content can be accessed using the **Add allowed forums** and **Add excluded forum** links.



Adding allowed forums

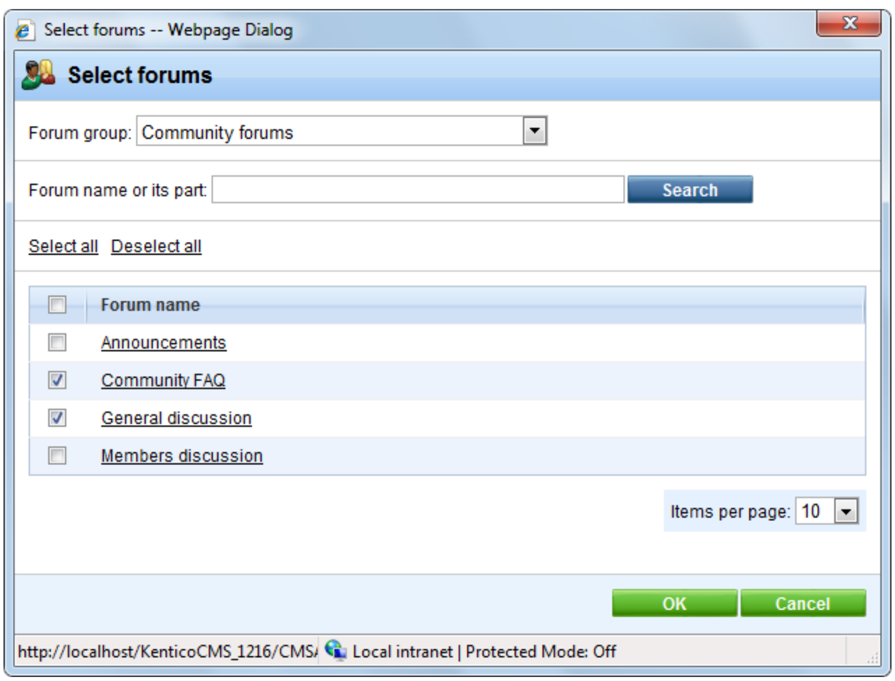
1. Click the **Add allowed forums** link.

2. In the following dialog, first use the **Site name** drop-down to choose the site whose forums will be indexed. If you select **(all)**, all forums on all sites in the system will be indexed.



3. If you selected a particular site in the previous step, click the **Select** button next to the **Forums** field. The dialog depicted in the screenshot below will be displayed.

Use the **Forum group** drop-down to select a forum group. Its child forums will be listed below. To include a forum in the index, enable ( ) the appropriate check-boxes. Click **OK** to save the settings.



Alternatively, the code names of forums, separated by semicolons, can be entered manually into the **Forums** field. All forums on the selected site can be added to the index this way, including group forums. The asterisk character (*) can be used as a wildcard for any number of characters.

For example, entering **community** would add all forums that contain the string *community* in their code name to the index.

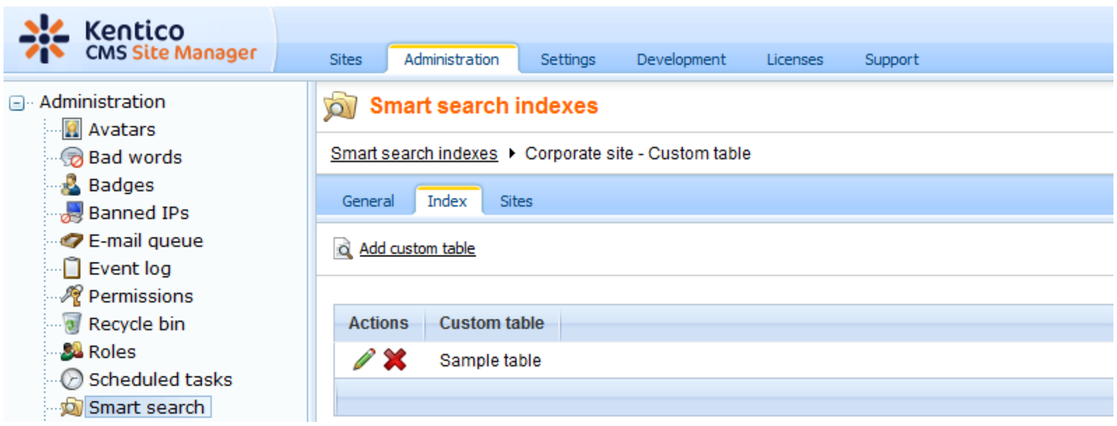
Adding excluded forums

Excluded forums make sense only when you have all forums defined as allowed. By defining a forum as excluded, it will not be indexed and all forums except for the excluded one will be indexed.

You can define an excluded forum using the **Add excluded forums** link, while the procedure is the same as when adding allowed forums.

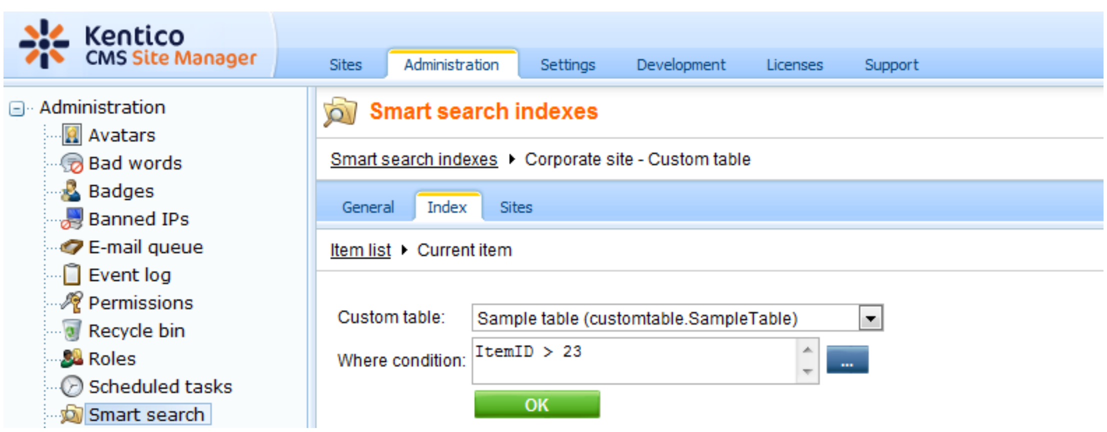
8.36.4.4 Defining custom tables index content

When editing () a custom table index on the **Index** tab in **Site Manager -> Administration -> Smart search**, you can see a list of custom tables indexed by the index. Custom tables can be added to the index using the **Add custom table** link. You can also **Edit** () the way listed custom tables are indexed or **Delete** () them from the list.



When adding a new custom table to the index or editing an existing one, you have the following options:

- **Custom table**: custom table to be indexed
- **Where condition**: WHERE part of the SELECT SQL query retrieving data from the table; using this, you can limit which items (rows) in the table will be indexed

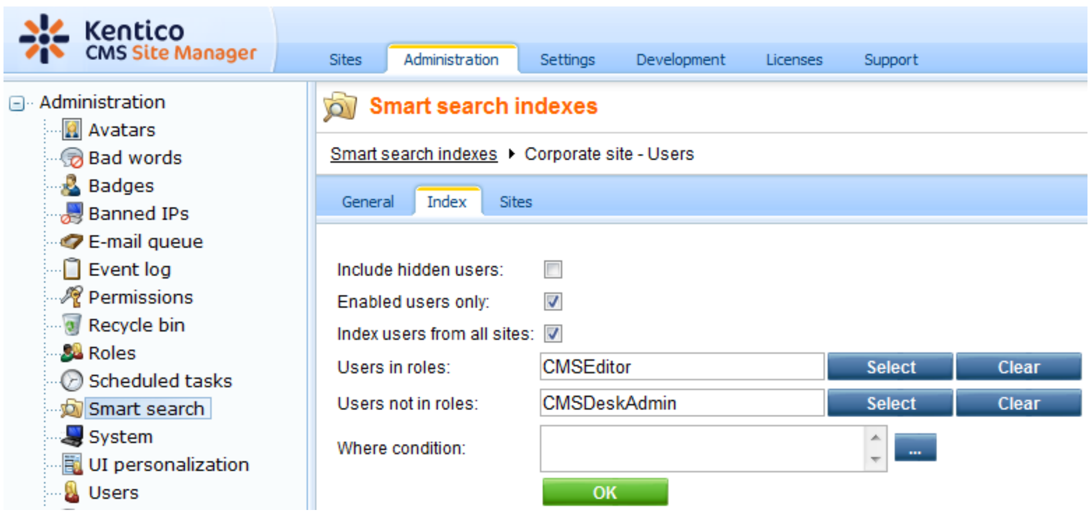


8.36.4.5 Defining user index content

When editing () a user index on the **Index** tab in **Site Manager -> Administration -> Smart search**, you can limit which users will be indexed. The following limitations can be set:

- **Include hidden users**: if enabled, hidden users will be indexed
- **Enabled users only**: if enabled, only enabled users will be indexed
- **Index users from all sites**: if enabled, users from all sites will be indexed; if disabled, only users from the sites assigned on the Sites tab will be indexed
- **Users in roles**: if entered, only users from the entered roles will be indexed
- **Users not in roles**: if entered, only users who are not in the entered roles will be indexed
- **Where condition**: WHERE part of the SELECT query on the *View_CMS_User* view used to retrieve users

Please note: you can define which fields will be indexed in **Site Manager -> Development -> System tables ->** edit () the **CMS_User** system table **-> Search fields**.



8.36.5 Settings for particular object types

The following text describes the possible settings for particular object types which can be indexed and subsequently searched.

Document types, Custom tables and Users

Settings for document types, custom tables and users are almost identical. You can find them on the **Search fields** tab in the editing interfaces for document types, custom tables and users.

In the top part of the tab, you can specify how the objects will be displayed in search results:

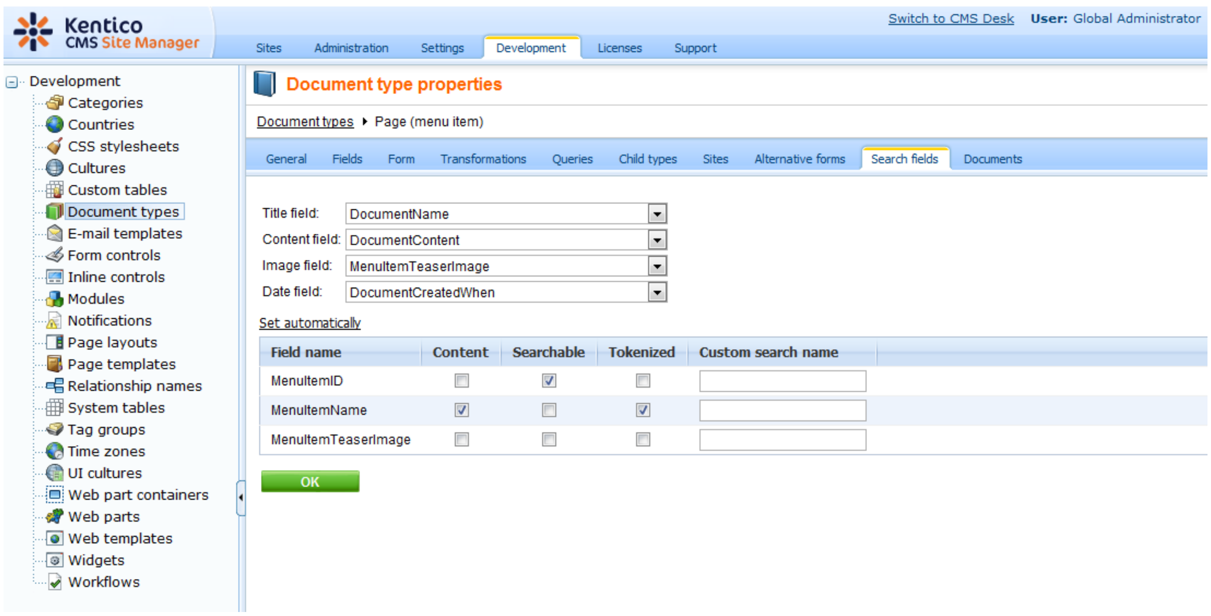
- **Title field** - specifies which field will be used as the title of the result item
- **Content field** - specifies which field will be used for the content extract of the result item
- **Image field** - specifies which field will be used for the image of the result item (not available for users, Avatar image is used for users by default)
- **Date field** - specifies which field will be used for the date and time displayed with the result item

Lines of the table in the bottom part of the page represent fields defined on the **Fields** tab, while columns define the following:

- **Content** - if checked, content of the field will be indexed and searchable the standard way
- **Searchable** - if checked, the content of the field will be searchable by entering an expression in

format: *<field code name>:<searched phrase>*. Please refer to the Search syntax topic for more information about field searches.

- **Tokenized** - indicates if the content of the field should be processed by the analyzer when indexing; the general rule is to use this for Content fields and not for Searchable fields
- **Custom search name** - relevant for Searchable fields; substitutes field code name in the <field code name>:<searched phrase> expression; if entered, the original code name can't be used



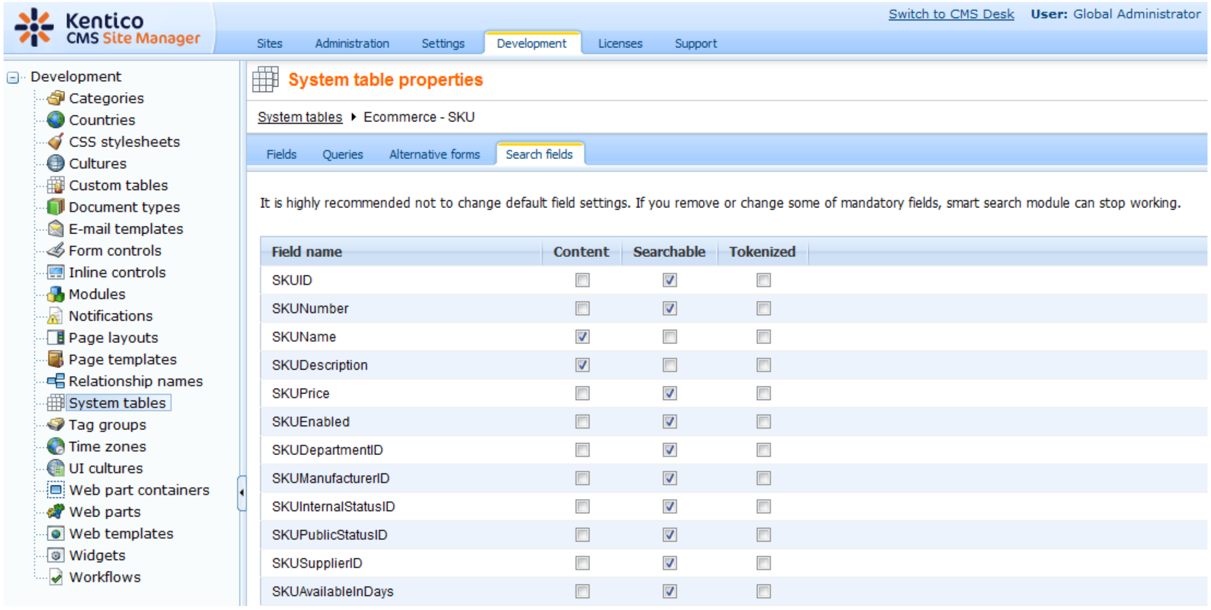
E-commerce SKUs

Similar settings are available for **E-commerce SKUs**. The settings can be done in **Site Manager -> Development -> System tables -> Edit () Ecommerce - SKU -> Search fields** tab. In this case, only the following settings are available:

- **Content** - if checked, content of the field will be indexed and searchable the standard way
- **Searchable** - if checked, content of the field will be searchable in format **<field code name>:<searched phrase>**, which searches only through the fields and not through the other fields
- **Tokenized** - indicates if the content of the field should be processed by the analyzer when indexing; the general rule is to use this for Content fields and not for Searchable fields

Important!

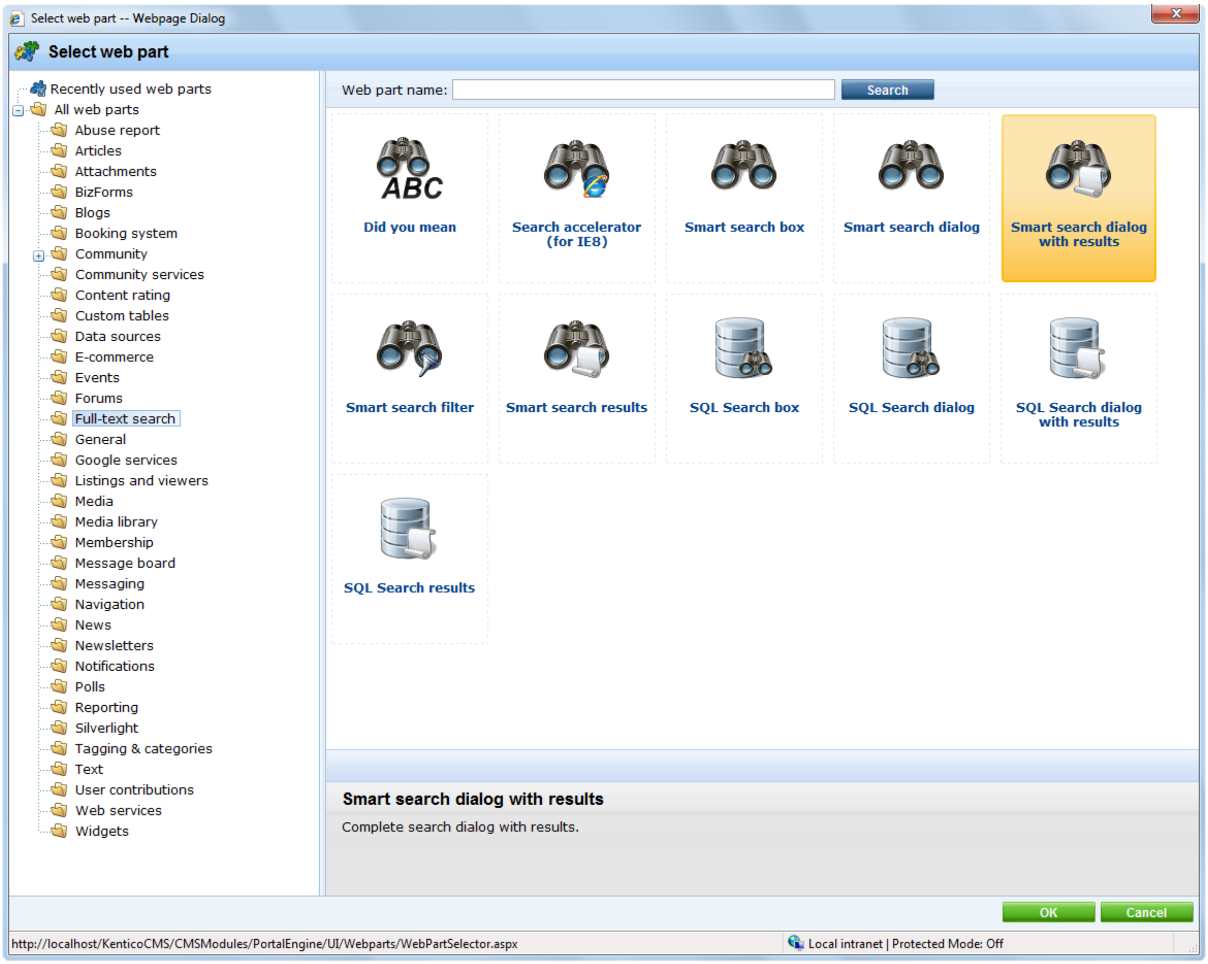
It is highly recommended that you do not modify settings other than those of your custom fields. If you modify the settings of some of the default fields, the functionality of searching through SKUs may get broken.



8.36.6 Available web parts

The Smart search module comes with the web parts mentioned in the text below. Only the most important web part properties are mentioned here. For a complete list and explanations of web part properties, please refer to [Kentico CMS Web Parts]() reference or click the **Documentation** link at the top right corner of the web part properties window.

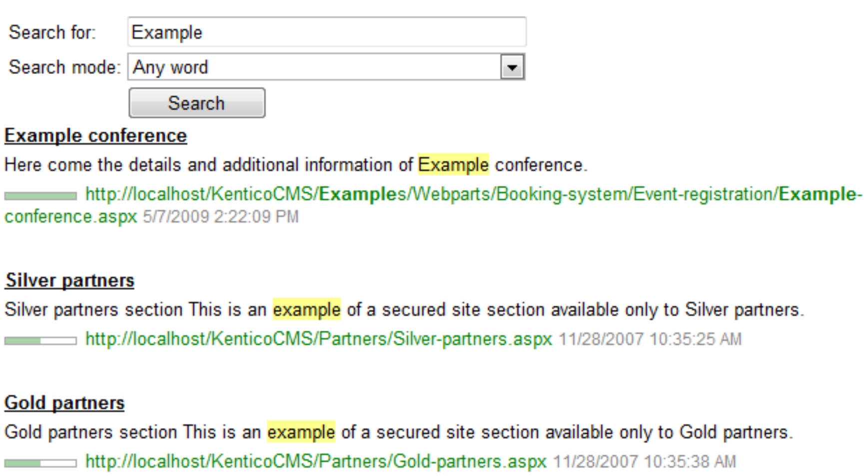
Smart search web parts are located in the **Full-text search** web part category.



The following web parts come with the Smart search module:

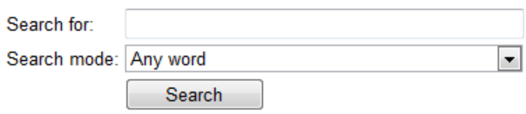
Smart search dialog with results

This is the all-in-one web part for both searching and displaying search results.



Smart search dialog

This web part needs to be placed on a page together with the **Smart search results** web part. The functionality of the two web parts will be identical to Smart search dialog with results, with the difference that these two web parts can be placed at different sections of the page. You can also connect the web part to a Smart search filter.

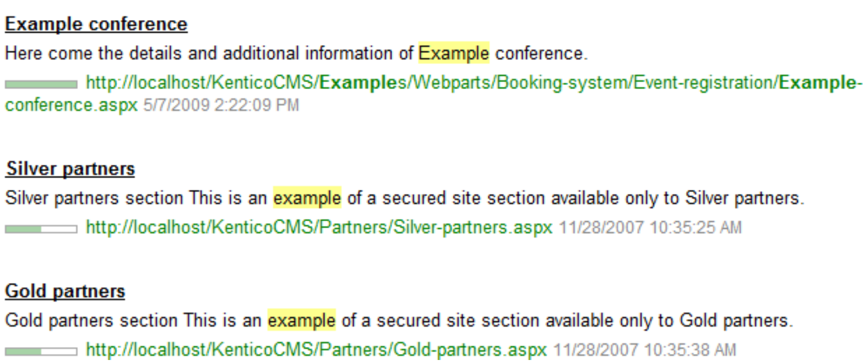


Smart search box

This web part is similar to the Smart search dialog, with the difference that users can not select the Search mode, which is hard-set in its web part properties. It is useful where limited space is available for the web part, e.g. in menus, etc.

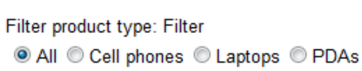
Smart search results

This web part can be used to display results of a search request sent from a **Smart search box** or **Smart search dialog** web part. It can be placed either on the same or on a different page as one of the two web parts. In case that it is placed on a different page, the page needs to be specified by their **Search results page URL** property.



Smart search filter

This web part can be used to enable users to search only documents of a specific type. You can find a detailed description of how this works in the Using the Smart search filter topic.



The following properties of **Smart search dialog with results** and **Smart search results** web parts are the most important:

Property Name	Description
Indexes	Determines which index will be searched; multiple indexes can be entered and then searched at the same time.
Transformation name	Name of the transformation used to display search results; there are two default transformations: **CMS.Root.SmartSearchResults** and **CMS.Root.SmartSearchResultsWithImages**.
Search options	Sets the level of conditions which users can enter: **None**: Users can specify only keywords, everything in the text box is escaped. **Basic**: Users can input special characters (such as +,-,...) but cannot filter by document field name (field:value). **Full**: All search options can be used. More information can be found in the Search syntax topic.
Search condition	Using this property, you can limit which documents will be searched using the search syntax (e.g. articleid:[(int)25 TO (int)150]).
Search results order	Defines sorting of the search results. You can specify one or more document fields (separated by commas). Search result will be sorted by them. You can input the ##SCORE## macro which means that results will be sorted by score. Default direction of sorting is ascending. You can change direction to descending by DESC keyword (e.g. articleid DESC). If you encounter the "field <fieldname> does not appear to be indexed" error when using multiple indexes, try specifying the type of the field in the following syntax: *(date)documentcreatedwhen*

8.36.7 Using the Smart search filter

The **Smart search filter** web part can be used to allow users to limit the range of objects that will be searched (conditional filter), or define how the search results will be sorted. It is designed to be connected to the **Smart search dialog** or **Search dialog with results**. You can connect more than one filter to these web parts.

You can see an example of how this works on the sample Corporate Site, on the **Examples -> Web parts -> Full-text search -> Smart search filter** page. You can see three **Smart search filters** connected to a **Smart search dialog**, which is connected to **Smart search results**.

The behaviour of the smart search filter is mainly defined by the properties explained in the table below. You can find descriptions of all the web part's properties in the Kentico CMS Web Parts reference or after clicking the **Documentation** link at the top right corner of the web part properties window.

Property Name	Description
Search webpart ID	ID of the Smart search dialog or Smart search dialog with results web part to which the filter should be connected.

Values	Using this property, you can specify the possible filtering options that can be selected. You enter one option per line in format: *<index field name>;<value of the field>;<displayed text>* When creating a conditional filter, the logical value of each filtering option can be specified. If the + symbol is added before the option, then only objects whose value in the given field matches the specified value will be included in the search. If the **-** symbol is added, only results that do **not** match the value will be returned. Please note, when entering integer or double type fields as filter options, the type of the value needs to be specified in the following way: *+DocumentCreatedByUserID;**(int)**53;Administrator* Examples: • ;;All • +classname;cms.cellphone;Cell phones • +_created;[{%CurrentTime|(add)-10080|(tosearchdatetime)%} TO {%CurrentTime|(tosearchdatetime)%}];Past week • ##SCORE##;;Score • documentcreatedwhen;;Creation date

Query name	Name of the query which can be used instead of the *Values* property to dynamically create the filter options. The query must return three columns, which will be used in the following order: *<index field name>, <value of the field>,<displayed text>*
	
	Sample query to load all document types as the filter options:
	
	`SELECT TOP 1000 '+classname', ClassName, ClassDisplayName`
	`FROM CMS_Class`
	`WHERE ClassIsDocumentType = 1`

Filter clause	Sets a clause that overrides the logical values specified for filtering options. Possible choices are: • **None** - no clause is added and the original logical values set for individual filtering options are used. • **Must** - indicates that the conditions in all filtering options must be fulfilled (adds the + symbol). • **Must not** - indicates that the conditions in all filtering options must not be fulfilled (adds the **-** symbol). Conditions are inverted compared to the **Must** option.
Filter is conditional	If true, the filter limits the searched documents (where condition). If false, the filter determines the sorting of search results (order by condition).

8.36.8 Search syntax

A detailed description of Lucene query parser syntax can be found at http://lucene.apache.org/java/2_1_0/queryparsersyntax.html.

Based on the level of allowed syntax specified by the **Search options** property of **Smart search dialog with results** or **Smart search results** web parts, users can enter search queries with restriction as described below:

- **Basic** - all Lucene search query syntax (as described on the page linked above) will be recognized *except for field searching*.
- **None** - no query syntax will be recognized. All text entered by users will be processed as a part of the searched for expression.
- **Full** - all search query syntax will be processed, including field searching.

Field search

By default, field searching can only be used to define an additional condition in a search expression. All conditions must start with either the + or - symbol. The + symbol indicates that only results which fulfill the field condition should be returned. The - symbol has the opposite meaning, only results that do **not** contain the specified value in the given field will be retrieved.

For example:

- network +NewsReleaseDate:[20080101 TO 20091231]

When searching for this expression using a document index, only news documents containing the word *network*, released in the year 2008 or 2009 will be returned.

If you wish to allow direct field searches, you must add the following key into the **/configuration/appSettings** section of your **web.config** file:

```
<add key="CMSSearchOnlyWhenContentPresent" value="false" />
```

Once this is done, expressions containing only search syntax can be used. This allows users to find records directly by entering an exact field value:

- *DocumentNodeID:(int)17* - the document with a *nodeID* equal to *17* will be returned.
- *NewsTitle:"New features"* - the news document titled *New features* will be returned.

Keep in mind that only fields that are set as **Searchable** in the field configuration of the given object type may be included in field searches. Information about the search field configuration of objects can be found in the Settings for particular object types topic.

Searching numeric fields

When performing field searches, the values specified are understood as strings by default. If you know that you are searching in **integer** or **double** fields, you will need to specify the type of the field the following way:

- NewsID:**(int)**22

- SKUPrice:**(double)**255.0
- DocumentNodeID:[**(int)**1 TO **(int)**100]

Searching date and time fields

Due to the same reason, search in **DateTime** fields also requires special syntax in format **<field name>:yyyymmddhhmm**. So for example:

- DocumentCreatedWhen:200812230101
- DocumentCreatedWhen:[200902020101 TO 200906020101]

If you need to specify date and time using a macro, you will need to use the **(tosearchdatetime)** parameter to convert the returned value to a format suitable for Lucene.

- {%CurrentDateTime|(tosearchdatetime)%}

You can also use the **(add)** parameter, which adds the specified amount of **minutes** to the value.

- {%CurrentDateTime|(add)-1440|(tosearchdatetime)%}

Field search with Stop and Simple analyzers

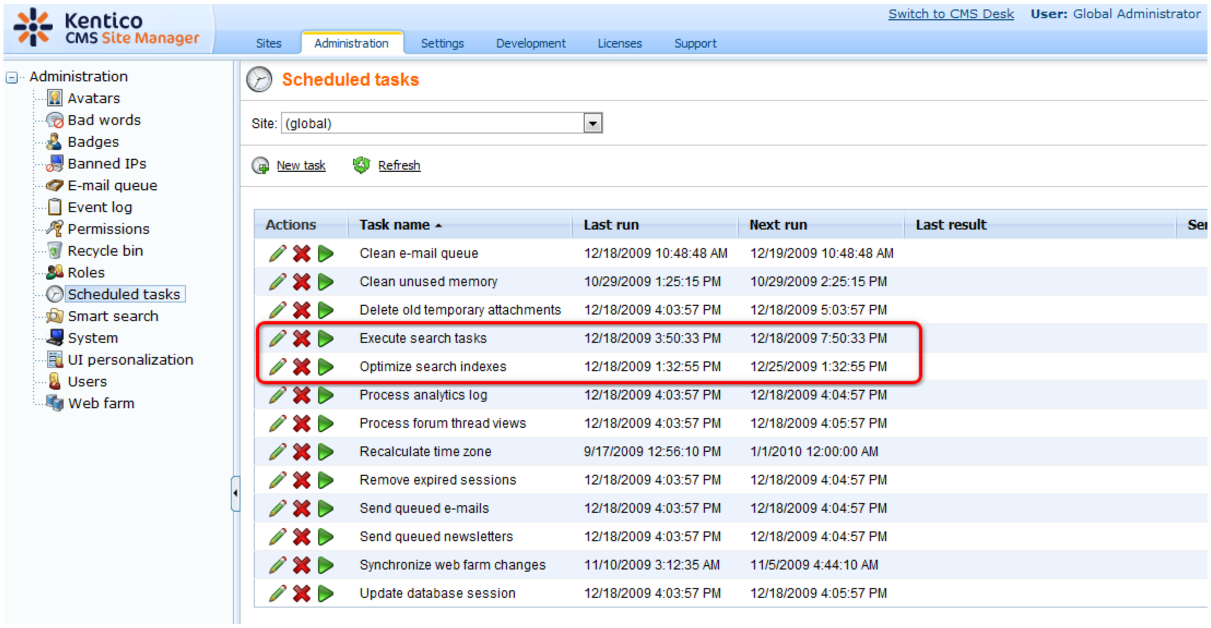
Indexes created by Stop and Simple analyzers cannot be searched using the standard field search format. This is by design, but you can use a workaround with a range query containing identical boundaries:

- newsid:[(int)22 TO (int)22]

8.36.9 Related scheduled tasks

The following two scheduled tasks are related to the Smart search module:

- **Optimize search indexes -** performs index optimization (defragmentation resulting in better performance, particularly in case of large indexes); by default once per week
- **Execute search tasks -** executes indexing tasks (created and executed automatically on content changes) that were not completed successfully on their automatic execution; by default every 4 hours



Please refer to the Development -> Scheduler chapter for more information about scheduled tasks.

8.36.10 Searching attachments

Document attachments uploaded to the database can be searched as well. This is done using the original SQL search. You need to have your SQL server configured properly as described in the Installation and deployment -> Additional configuration tasks -> Configuration of full-text search in files chapter in order for attachment searching to be functional.

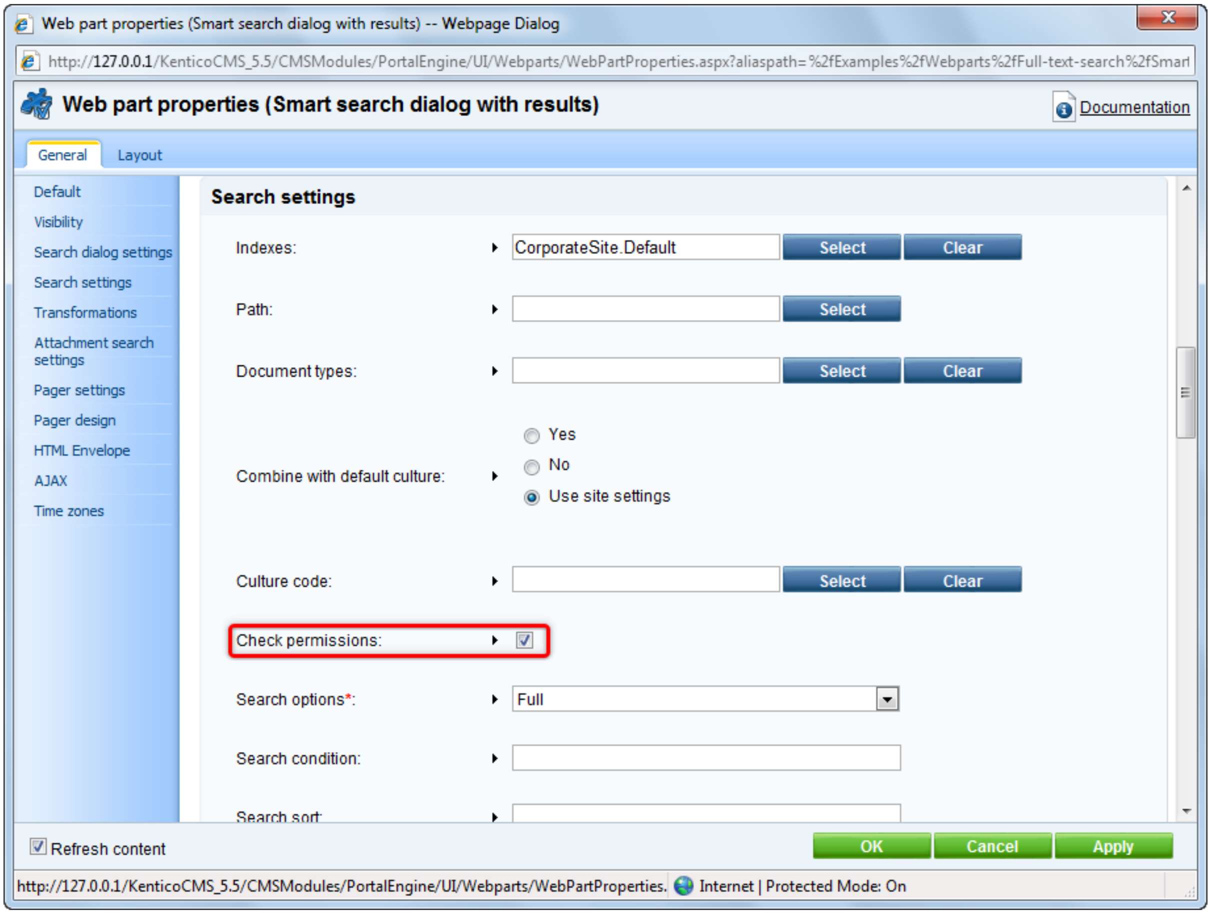
If you have the SQL server configured, you need to enable attachment searching by using the following properties of the **Smart search dialog with results** or **Smart search results** web part:

Property Name	Description
Search in attachments	If enabled, document attachments will be searched.
WHERE condition	WHERE condition specifying documents whose attachments will be searched.
ORDER BY expression	ORDER BY expression ordering documents. The attachments will be displayed according to this order in the search results.

The results are interlaced with regular search results, which is by design and cannot be modified.

8.36.11 Security

Smart search web parts have the **Check permissions** property. If this option is enabled, pages for which the user does not have permissions will not be displayed to them in the search results.



8.36.12 SQL Search overview

The SQL Search engine is kept in the system for backward compatibility with older versions and for attachment searching. It uses standard SQL queries to search for the given expression. It runs a separate query for each document type. You can find the search query for a document type at **Site Manager -> Development -> Document types -> ... select document type ... -> Queries -> Edit** ( ) the **searchtree** query.

Common fields (such as document name) are searched using the **cms.root.searchdocuments** query.

Uploaded files are searched using the **cms.root.searchattachments** query. If you want to search uploaded files, you need to configure the system as described in the [Installation and deployment -> Additional configuration tasks -> Configuration of full-text search in files]() topic. In this case, the files are searched using the Microsoft SQL Server Search Engine.

Web parts and controls

The search dialog can be easily integrated into the website using the web parts in the Full-text search category (SQL search dialog, SQL search box, etc.) or using the CMSSearchDialog and CMSSearchResults server controls (in your ASP.NET code).

Excluding documents from search

You can **exclude document types** by setting the **Exclude document types from SQL search** value in **Site Manager -> Settings -> Web site** dialog. You need to enter the code name, such as cms.article. You can enter multiple document types separated with a semicolon (;).

You can **exclude website sections** from search by setting the **Exclude documents from SQL search** in **Site Manager -> Settings -> Web site** dialog. You need to enter the alias path of the section that should be excluded. A single document can be excluded using e.g. **/news/news1**, whole website section can be excluded using e.g. **/news/%**. You can enter multiple values separated by semicolons (;).

Modifying the search result format

If you only need do modify the search results format, you can do that by modifying the transformation **searchresults** in **Site Manager -> Development -> Document Types -> edit ( ) Root -> Transformations**.

Development of Custom Search Provider

If you need to integrate a custom search engine or make additional modifications to the search results returned by the standard search engine, you can develop your own search provider. You can find more details in the API programming and Kentico CMS internals-> Custom providers -> Custom Search Provider topic.

8.36.13 Smart search internals and API

8.36.13.1 Database tables and API classes

The Smart search module uses the following database tables:

- **CMS_SearchIndex** - contains smart search indexes and their settings
- **CMS_SearchIndexCulture** - assigns smart search indexes to the cultures they should index
- **CMS_SearchIndexSite -** assigns smart search indexes to the sites they should index
- **CMS_SearchTask -** contains smart search indexing tasks

The Smart search API is provided by the following **CMS.SiteProvider** namespace classes:

- **SearchIndexInfo**, **SearchIndexInfoProvider** - these classes provide functionality for managing smart search indexes
- **SearchTaskInfo**, **SearchTaskInfoProvider -** these classes provide functionality for managing smart search indexing tasks

As well as the following **CMS.SettingsProvider** namespace classes:

- **SearchIndexSettings**, **SearchIndexSettingsInfo -** these classes provide functionality for managing smart search index settings

The following topics show examples of how these classes can be used:

- Getting search index data
- Managing search indexes
- Performing a search
- Creating an indexing task

The following topic shows how search results can be displayed in transformations:

- Search results in transformations

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.36.13.2 Getting search index data

The following sample code shows how you can get a search index as a *SearchIndexInfo* object:

[C#]

```
using CMS.SiteProvider;

...

        // Get search index object from database by searchIndexID
        SearchIndexInfo sii = SearchIndexInfoProvider.GetSearchIndexInfo(1);
```

The following sample code shows how you can get a DataSet containing search indexes from the system:

[C#]

```
using System.Data;
using CMS.SiteProvider;

...

        string where = "IndexID < 10";
        string orderby = "IndexDisplayName";
        int topN = 10;
        string selectedColumns = "IndexID, IndexName, IndexSettings, IndexType";

        // Get dataset of search indexes according to the given parameters
        DataSet ds = SearchIndexInfoProvider.GetSearchIndexes(where, orderby);

        DataSet ds2 = SearchIndexInfoProvider.GetSearchIndexes(where, orderby, topN,
```

8.36.13.3 Managing search indexes

The following sample code shows how you can modify the *Display name* of an existing search index:

[C#]

```
using CMS.SiteProvider;

...

        // Get index object from database by index code name
        SearchIndexInfo sii = SearchIndexInfoProvider.GetSearchIndexInfo("TestIndex"

        // If index exists
        if (sii != null)
        {
            // Update some property
            sii.IndexDisplayName += " updated";

            // Save to database
            SearchIndexInfoProvider.SetSearchIndexInfo(sii);
        }
```

The following sample code shows how you can create a new search index, assign a site and culture to it and add it to the system:

[C#]

```
using System;
using CMS.SiteProvider;
using CMS.SettingsProvider;
using CMS.CMSHelper;

...

        // Create empty index object
        SearchIndexInfo sii = new SearchIndexInfo();

        // Set properties
        sii.IndexIsCommunityGroup = false;
        sii.IndexName = "TestIndex";
        sii.IndexDisplayName = "Test index";
        sii.IndexType = PredefinedObjectType.DOCUMENT;
        sii.IndexAnalyzerType = AnalyzerTypeEnum.StandardAnalyzer;
        sii.StopWordsFile = "";
        sii.CustomAnalyzerName = "";

        // Create new index settings
        SearchIndexSettingsInfo sisi = new SearchIndexSettingsInfo();

        // Set setting properties
        sisi.IncludeBlogs = true;
        sisi.IncludeForums = true;
        sisi.IncludeMessageCommunication = true;
        sisi.ClassNames = ""; // for all document types
        sisi.Path = "/%";
        sisi.Type = SearchIndexSettingsInfo.TYPE_ALLOWED;
```

```
        sisi.ID = Guid.NewGuid();

        // Save index settings
        SearchIndexSettings sis = new SearchIndexSettings();
        sis.SetSearchIndexSettingsInfo(sisi);
        sii.IndexSettings = sis;

        // Save to database
        SearchIndexInfoProvider.SetSearchIndexInfo(sii);

        // Add current culture to index
        SearchIndexCultureInfoProvider.AddSearchIndexCulture(sii.IndexID,
CMSContext.CurrentDocumentCulture.CultureID);

        // Add current site to index
        SearchIndexSiteInfoProvider.AddSearchIndexToSite(sii.IndexID, CMSContext.Cur
```

The following sample code shows how you can rebuild an existing search index:

[C#]

```
using CMS.SiteProvider;

...

        // Get index object from database by index code name
        SearchIndexInfo sii = SearchIndexInfoProvider.GetSearchIndexInfo("TestIndex"

        // If index exists
        if (sii != null)
        {
            // Create rebuild task
            SearchTaskInfoProvider.CreateTask(SearchTaskTypeEnum.Rebuild, sii.IndexT
        }
```

The following sample code shows how you can delete an existing search index:

[C#]

```
using CMS.SiteProvider;

...

        // Get index object from database
        SearchIndexInfo sii = SearchIndexInfoProvider.GetSearchIndexInfo("TestIndex"

        // If index exists
        if (sii != null)
        {
            // Delete from database
```

```
            SearchIndexInfoProvider.DeleteSearchIndexInfo(sii);
        }
```

8.36.13.4 Performing a search

The following sample code shows how you can perform a search and get the results into a DataSet:

[C#]

```
using System.Data;
using CMS.SiteProvider;
using CMS.CMSHelper;
using CMS.GlobalHelper;

...

        // Get index object from database by index code name
        SearchIndexInfo sii = SearchIndexInfoProvider.GetSearchIndexInfo("TestIndex"

        int numberOfResults;

        // If index exists
        if (sii != null)
        {

            // Set properties
            string searchText = "Search text";
            string path = "/%";
            string classNames = "";
            string cultureCode = "EN-US";
            string defaultCulture = CultureHelper.DefaultCulture.IetfLanguageTag;
            Lucene.Net.Search.Sort sort = SearchHelper.GetSort("##SCORE##");
            bool combineWithDefaultCulture = false;
            bool checkPermissions = false;
            bool searchInAttachments = false;
            string searchIndexes = sii.IndexName;
            int displayResults = 100;
            int startingPosition = 0;
            int numberOfProcessedResults = 100;
            UserInfo userInfo = CMSContext.CurrentUser;
            string attachmentWhere = "";
            string attachmentOrderBy = "";

            // Get search results
            DataSet ds = SearchHelper.Search(searchText, sort, path, classNames,
cultureCode, defaultCulture, combineWithDefaultCulture, checkPermissions,
searchInAttachments, searchIndexes, displayResults, startingPosition,
numberOfProcessedResults, userInfo, out numberOfResults, attachmentWhere,
attachmentOrderBy);

            // If at least one item is found
            if (numberOfResults > 0)
```

```
            {
                // Perform some action with result dataset
            }
        }
```

8.36.13.5 Creating an indexing task

The following sample code shows how you can create a task to update a search index:

[C#]

```
using CMS.SiteProvider;
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.TreeEngine;

...

        string siteName = CMSContext.CurrentSiteName;
        string aliasPath = "/News";
        string cultureCode = "En-us";

        // Tree provider
        TreeProvider provider = new CMS.TreeEngine.TreeProvider(CMSContext.CurrentUs
        // Get document of specified site, aliaspath and culture
        TreeNode node = provider.SelectSingleNode(siteName, aliasPath, cultureCode);

        // If node exists
        if (node != null)
        {
            // Edit and save document node
            node.DocumentName += " changed";
            node.Update();

            // Create update task
            SearchTaskInfoProvider.CreateTask(SearchTaskTypeEnum.Update,
PredefinedObjectType.DOCUMENT, SearchHelper.ID_FIELD, node.GetSearchID());
        }
```

8.36.13.6 Search results in transformations

You can use the following default transformations to display search results using the **Smart search dialog with results** and **Smart search results** web parts:

- **CMS.Root.SmartSearchResults**
- **CMS.Root.SmartSearchResultsWithImages**

Search results are returned in a so-called **search dataset**. No matter what the field names in the found documents are, the search dataset always contains the following fields that are mapped to the real documents' fields:

score	Expresses the relevance of the found document in a numeric value ranging from 0 to 1, where higher values indicate higher relevance.
title	This field is mapped to the field specified by the **Title field** drop-down list on the **Search fields** tab of the particular document type.
content	This field is mapped to the field specified by the **Content field** drop-down list on the **Search fields** tab of the particular document type.
created	This field is mapped to the field specified by the **Date field** drop-down list on the **Search fields** tab of the particular document type.
image	This field is mapped to the field specified by the **Image field** drop-down list on the **Search fields** tab of the particular document type.

In transformations, you can get the values from the dataset by using the **Eval("<field name>")** function: Eval("score"), Eval("title"), ...

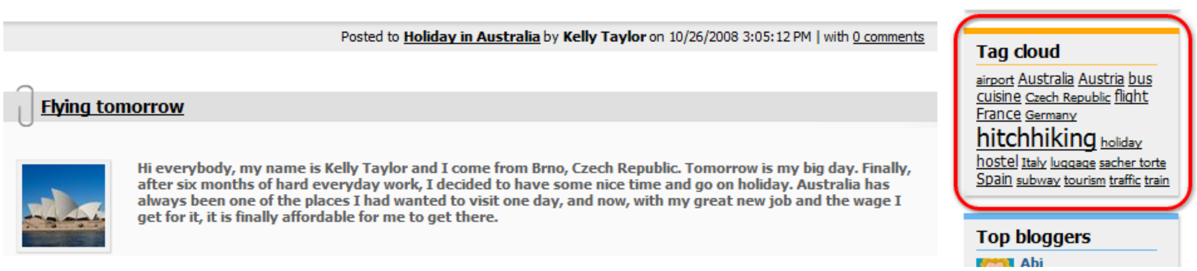
You can also use the following methods in your transformations:

- **SearchResultUrl(bool absolute) -** returns the URL to the document; the parameter indicates if the returned URL should be absolute
- **SearchHighlight(string text, string startTag, string endTag) -** wraps the text entered in the first attribute by the tags specified by the other two attribute
- **GetSearchImageUrl(string noImageUrl, int maxSideSize) -** returns the URL of the current search result image; the first attribute specifies the URL returned if no image is found, the second one specifies the maximal side size to which the image will be resized
- **GetSearchValue(string columnName) -** returns the value of the specified column for the current search result

8.37 Tags

8.37.1 Overview

The Tags module enables users to tag documents with key words, called tags, depending on the content of a particular document. Tags are associated with a document and are convenient for simple marking of documents according to various criteria, e.g. your interests.



The **Tags** module functions in a similar way like the Categories module as tags are organized in **tag groups**. However, categories can be defined by both the global administrator in Site Manager and user in CMS Desk whereas tag groups can only be defined by the global administrator. Besides, user-defined categories are available only to the user who created them.

Tag groups are related to a particular site as defined by the global administrator in Site Manager.

In order for the user to be able to tag documents, at least one tag group must be defined for the system by the global administrator. That is why the default tag group **Content** is available for all Kentico CMS sample sites, which is inherited and should not under standard circumstances be deleted from the system.

All tags created in CMS Desk by the user are available globally and can be used by any other user.

The Tags module comes with the **Tag cloud** web part that can display all tags within one tag group. The tags are displayed in the form of links by the web part. Clicking a link in the tag cloud redirects you to another page, where a repeater displays a list of all documents tagged with the given tag.

- To learn how to create, alter or delete a tag group, please refer to the Managing tag groups topic.
- To learn how to tag individual documents, please refer to the Tagging documents topic.
- To learn how to display documents associated with the selected tag, please refer to the Using the Tag cloud web part topic.

Kentico CMS Community Site Guide contains another example of the use of **Tag groups** as a part of the Tags module. You can find a step-by-step tutorial on how to create tag groups in Community Site Guide -> Part 2 -> Pre-development tasks -> Creating the tag groups.

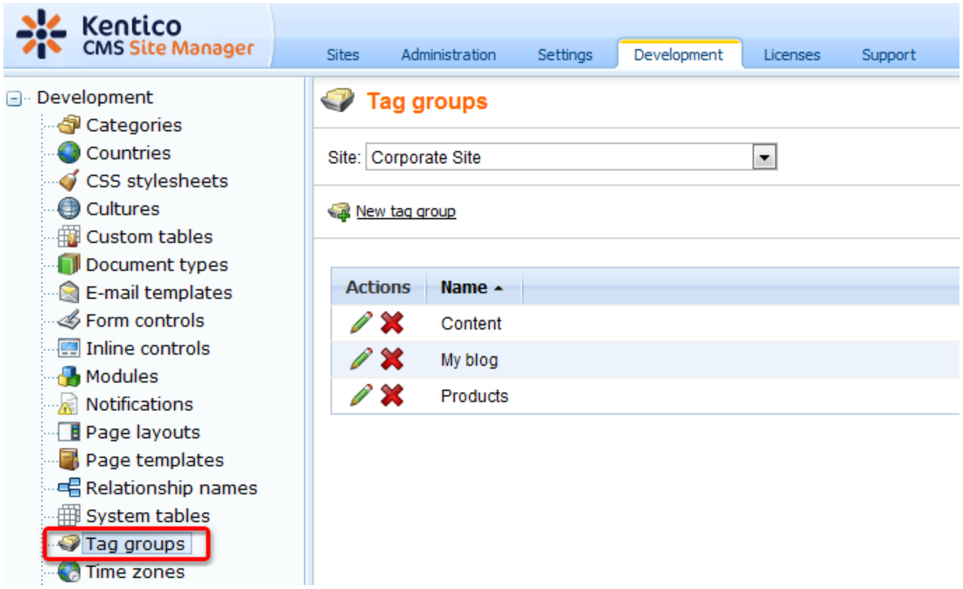
Several more examples of the use of the Tags module are available in Kentico CMS Community Site Guide; please note that these practical examples do not concern the whole functionality of the module but focus on its use in a broader context of the Community Site sample website:

- See Creating the Blogs section -> Creating the Blogs page in the same section of Kentico CMS Community Site Guide: A concise example of the use of the **Tag cloud** web part in the context of blogs.
- See Creating the Blog posts page: A further example of the use of the **Tag cloud** web part in the context of blogs.
- See Creating the News page: A concise example of the use of the **Tag cloud** web part in the context of news.

You will need to have the Community Site sample website installed to follow Kentico CMS Community Site Guide.

8.37.2 Managing tag groups

Tags are divided into tag groups, which are topic-related groups of tags. No global tag groups are defined and each tag group is bound to a particular site. Tag groups can be created and managed by the global administrator in **Site Manager -> Development -> Tag groups**. You can **Edit** ( ) or **Delete** ( ) the listed tag groups in this section.



Please note

At least one tag group must be defined in order to tag documents.

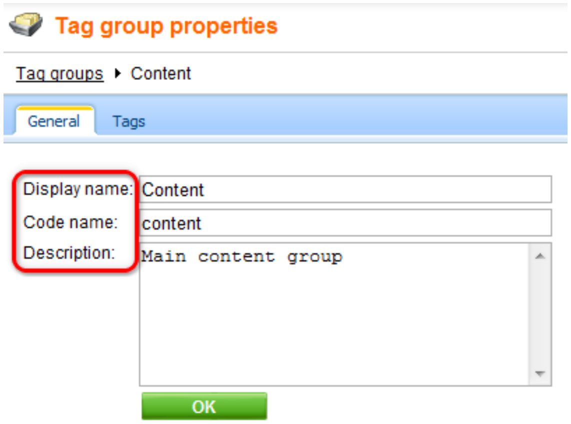
The default tag group **Content** is available for all Kentico CMS sample sites. It is inherited and should not under standard circumstances be deleted from the system.

Creating tag groups

In this example, you will learn how to create a new tag group.

1. To create a new tag group, go to **Site Manager -> Development -> Tag groups** and click the **New tag group** link. The following properties will have to be entered:

- **Display name** - display name of the tag group
- **Code name** - code name of the tag group
- **Description** - description text of the tag group



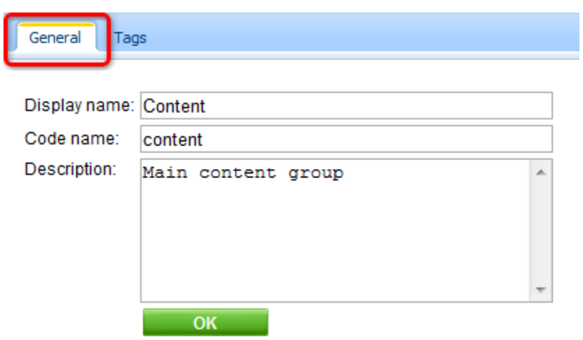
2. Click **OK** to confirm the values you have entered. The tag group will be created and become visible in the list.

Editing tag groups

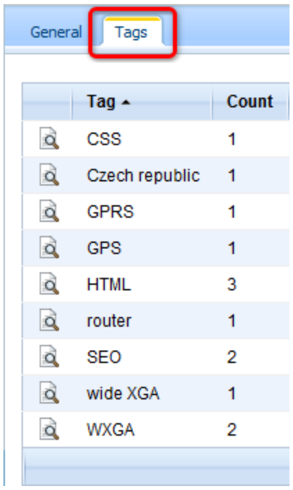
In this example, you will learn how to edit an existing tag group.

1. To edit an existing tag group, go to **Site Manager -> Development -> Tag groups** and choose to **Edit** ( ) one of the listed tag groups.

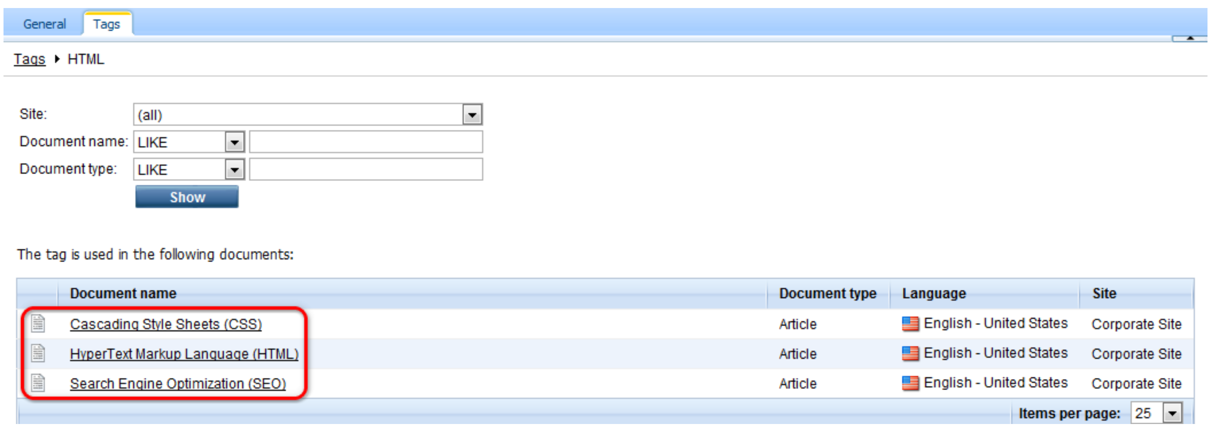
2. If you would like to change some of its properties, go to the **General** tab and change either the **Display name**, **Code name** or **Description** of the tag group.



3. If you would like to display the list of all tags in the selected tag group, switch to the **Tags** tab:



4. If you would like to display a list of links to documents tagged with a tag, click the icon next to the particular tag.



Please note

The global administrator cannot delete a tag from the document in **Site Manager**. This must be done by the user in **CMS Desk**.

8.37.3 Tagging documents

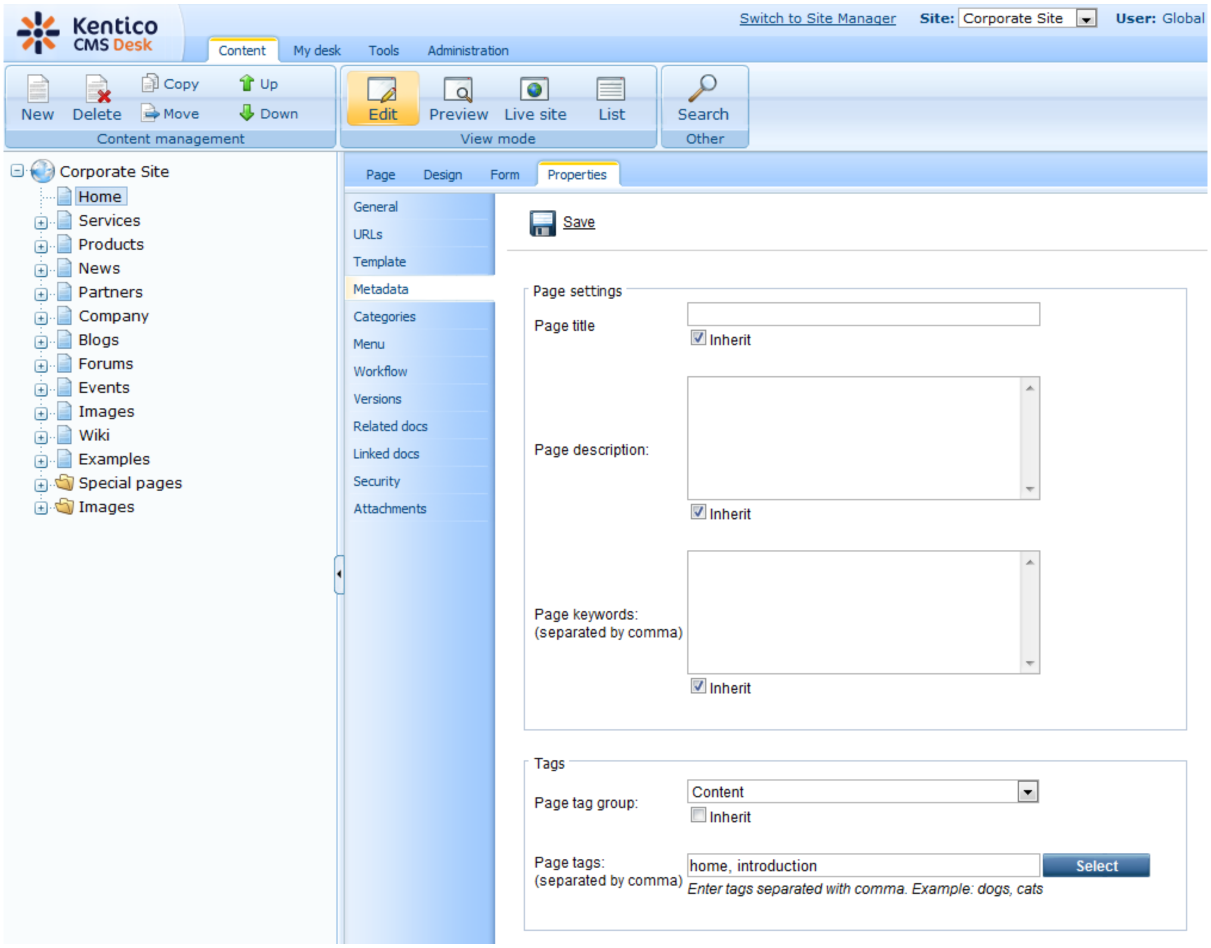
If you need to learn how to tag a document, the following examples describe this process. There are two ways of tagging documents:

- You can add tags to a document either by using the **Properties** tab of the **CMS Desk -> Content -> Edit** interface (as described here).
- Or you can use the **Fields** tab in **Site Manager -> Development -> Document types -> Edit ( ) <*document_type*>** (as described here).

Tagging documents using Properties

1. Go to **CMS Desk**, switch to the **Edit** mode and select the document that you wish to tag from the content tree.

2. Switch to its **Properties -> Metadata** tab.



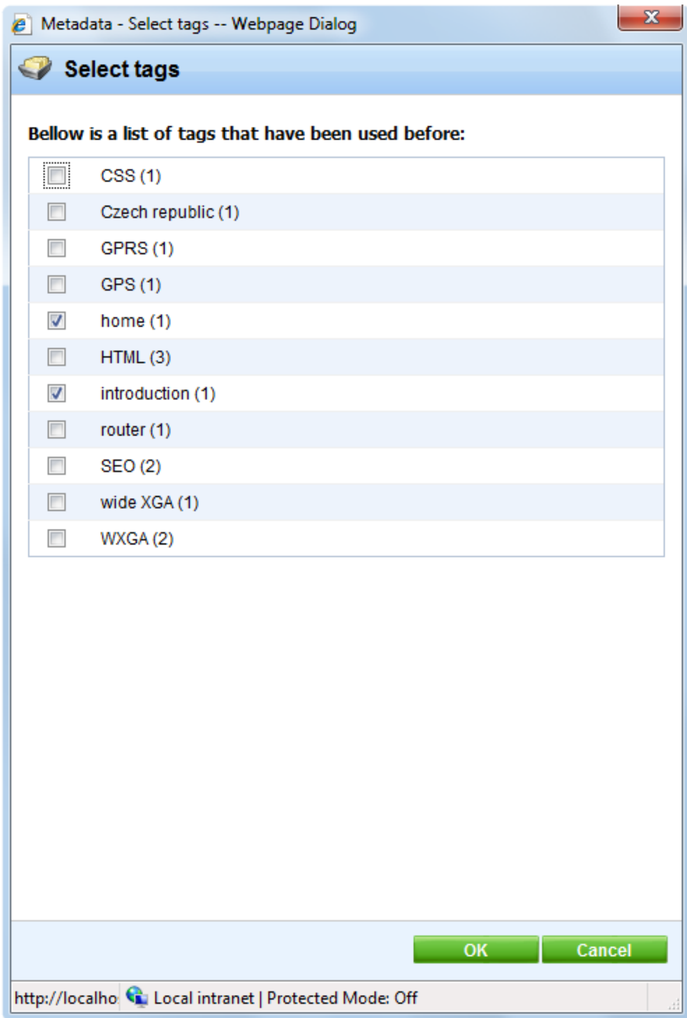
3. Choose a tag group using the **Page tag group** drop-down list. If you check the **Inherit** check-box, tag group will be inherited from the document's parent.

Please note

There needs to be **at least one tag group defined** for you to be able to tag a document. If there are no tag groups created, you will not be able to tag a document as each tag needs to belong to some tag group. You can learn about creating tag groups in the next chapter.

It is also recommended to select a tag group for the root of the content tree so that it could be inherited by the pages located under it.

4. Enter the tags into the **Page tags** field. Tags can be entered either manually or can be selected using the **Select** button. This button displays a list of all tags in the selected tag group. Check the appropriate tags' check-boxes and click **OK**.



5. Click **Save**. The entered tags will be saved and attached to the document.

Tag formats

When entering more than one tag into the **Page tags** field, the tags should be separated with a comma or a blank space. A combination of both in a single entry is also valid. The following examples are all valid entries for adding three tags - *tag1*, *tag2* and *tag3*:

tag1, tag2, tag3
tag1 tag2 tag3
tag1, tag2 tag3

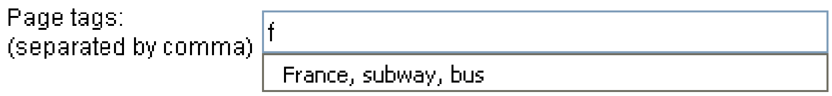
In case that you are entering a tag consisting of more than one word, you should enclose it within quotation marks. Multiple long tags can also be entered and can be also divided by both blank spaces and commas:

"long tag1", tag, "long tag2"
"long tag1" tag "long tag2"

Quotation marks can also be used for tags containing special characters that couldn't be used otherwise:

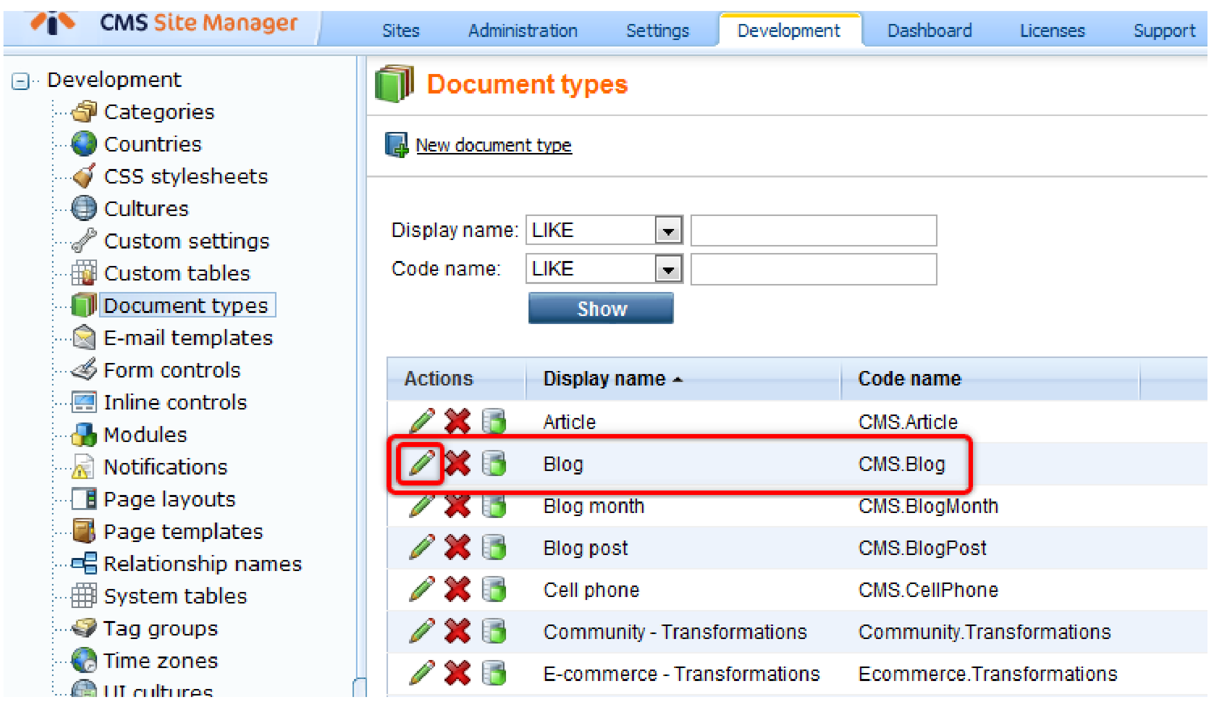
"tag@1", "tag#2", "long, strange: tag@#$"

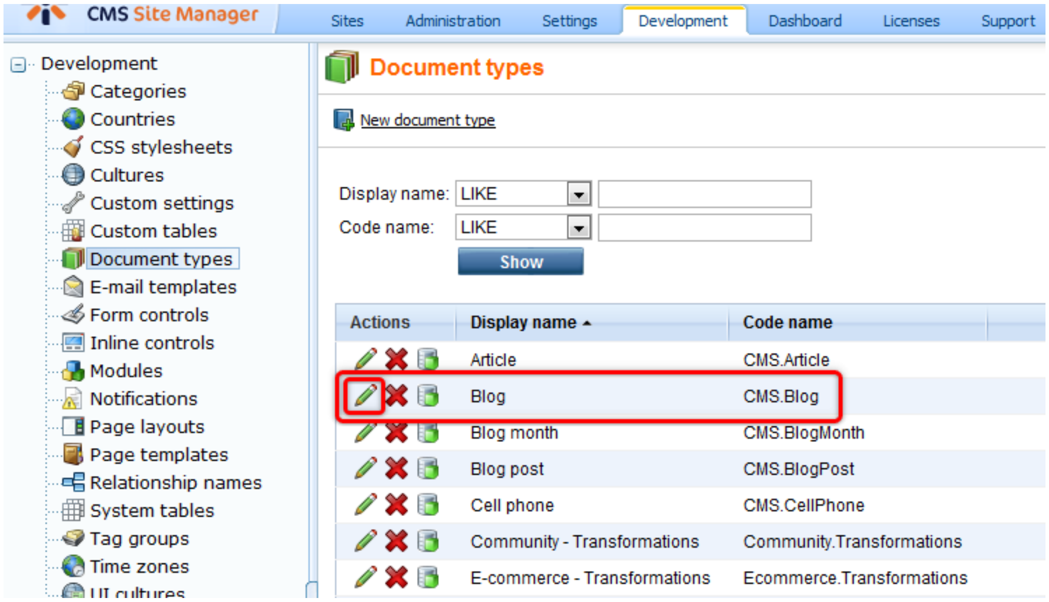
The page tags field has also an insinuation function implemented. This functions offers you tags from the selected tag group while you are writing:



Tagging documents using Fields

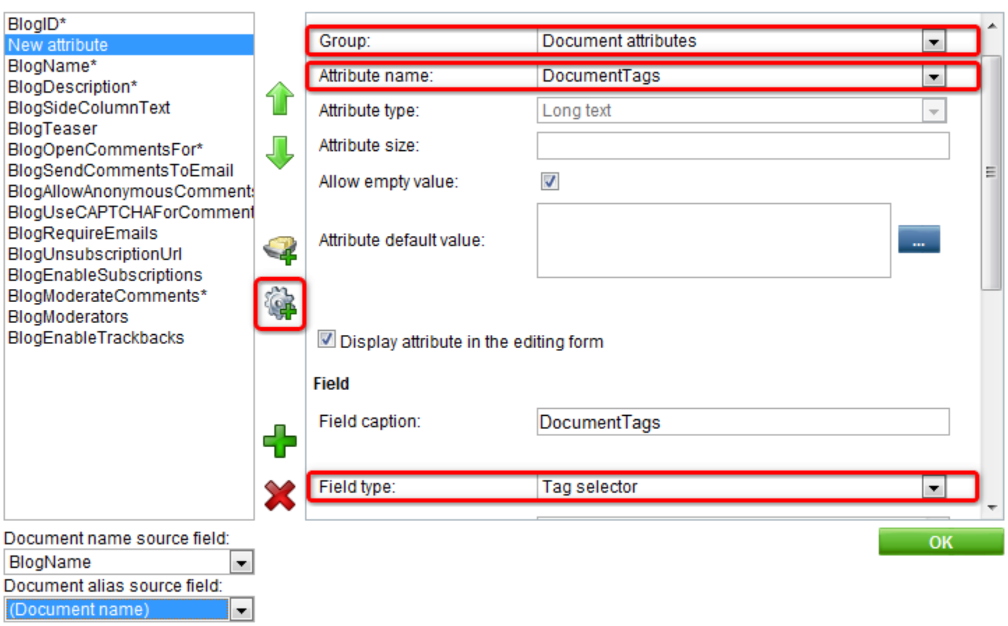
1. Go to **Site Manager -> Development -> Document types** and choose to **Edit** ( ) a document type.





2. Switch to the **Properties** tab, click the **New system attribute** () icon to create a new system attribute and from the corresponding drop-down lists select the following values:

- **Group:** Document attributes
- **Attribute name:** DocumentTags
- **Field type:** Tag selector



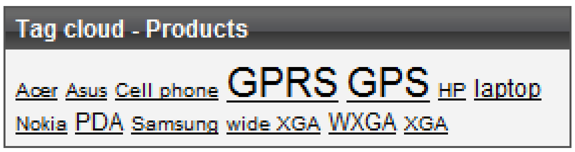
Click **OK** to save the changes.

Please note

> If you need to create a custom document type with a tag selector, the tag selector must be bound to a system field.

8.37.4 Using the Tag cloud web part

The **Tag cloud** web part is used to display tags that are associated with the current document. Tags are displayed in the form of links and those with higher occurrence are displayed in higher font size. If the site visitor clicks on some of the links, a list of all documents tagged with the given tag will be displayed.



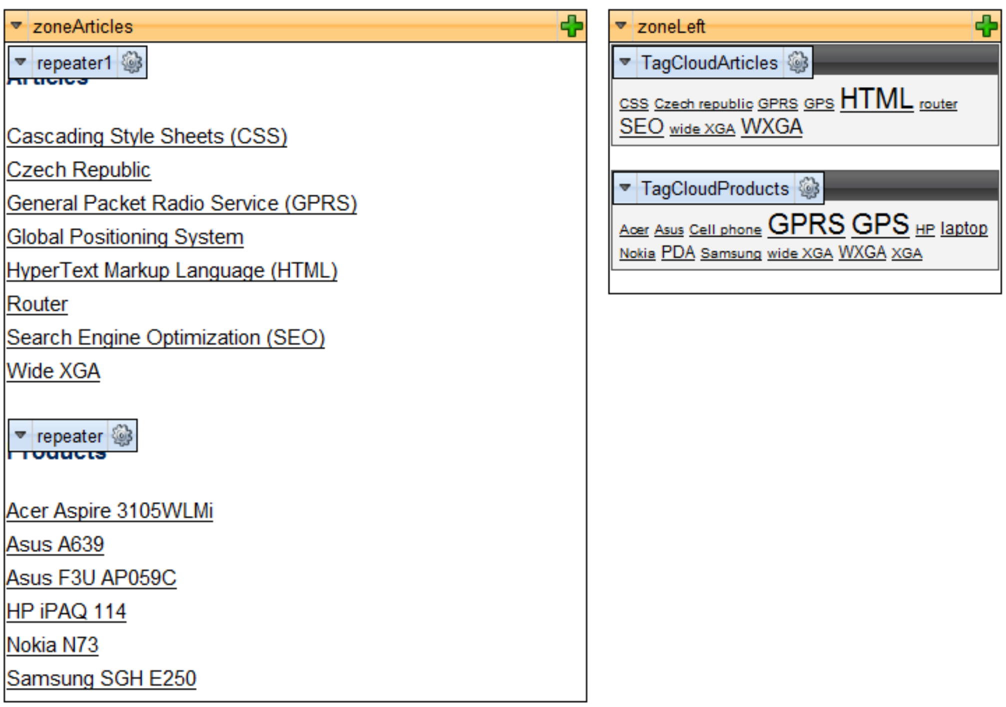
The web part has the following specific properties:

Tag filter	
Site name	Code name of the website from which you want to display the tags. If you leave the value empty, the content is retrieved from the current website.
Tag group name	Name of the tag group from which the tags will be displayed.
Select top N tags	Number of tags which will be displayed. Top tags according to the order specified by the *ORDER BY expression* property will be displayed.
ORDER BY expression	ORDER BY part of the SQL query used to retrieve the tags.
Tag cloud settings	
Document list URL	URL of the page on which the documents list will be displayed after clicking some of the tags (see below for an example).
Query string parameter name	Name of the parameter by means of which the tag ID will be transferred.
Tag separator	Separator that will be placed between tags in the tag cloud.
Minimal tag font size	Size of tags with the lowest occurrence.
Maximal tag font size	Size of tags with the highest occurrence.
Document filter	
Use document filter	Indicates if the document filter will be used. If true, the web part will display only tags used in the documents specified by the properties below.
Path	Path of the documents the tags of which will be displayed by the web part.
Combine with default culture	Indicates if tags from the default language version of the document

	should be displayed if the document is not translated to the current language.
Culture code	Culture version of the displayed tags.
Maximum nesting level	Maximum nesting level. It specifies the number of sub-levels in the content tree of the document specified by the *Path* property that should be included in the displayed content (i.e. whose tags will be displayed).
Select only published	Indicates if only tags from published documents should be displayed. Applied only when 'Site name' or 'Alias path' property is defined. If disabled, all tags from the selected tag group will be displayed.
Where condition	WHERE part of the SQL query used to retrieve tags.

Common usage

The Tag cloud is designed to "work in pair" with some repeater web part. When a tag link is clicked in the tag cloud, a list of all documents tagged with this tag is displayed in the repeater. The repeater can be placed either on the same page as the tag cloud or on some other page to that the site visitor will be redirected. You can see an example of this behavior on the Corporate Site sample website under **Examples -> Web parts -> Tagging & Categories -> Tag cloud**.



In order for the two web parts to cooperate correctly, you have to correctly set some of their properties:

Tag cloud:

- The placement of the repeater is defined by the Tag cloud's **Document list URL** property. In case that the repeater is placed on the same page as the tag cloud, the value should be left blank. In case that it is placed on some other page, you should enter the **alias path** of that page.

- ID of the clicked tag is transferred to the repeater in form of a query string parameter. The name of the parameter can be set using the **Query string parameter name** property. The repeater displays the appropriate list of documents based on the value that it gets by via this parameter.

Repeater:

- Set the value of the **Path** parameter to the location in the content tree where the documents are stored.

- Set the value of the **Document types** parameter to the document type(s) that is (are) to be displayed.

- Select the transformations that you want to use for the **Transformation** and **Selected items transformation**.

- Finally, use the following code as a value for the repeater's **WHERE condition** parameter. The **tagid** value should be replaced by the name set in the Tag cloud's **Query string parameter name**:

```
({?tagid|(toint)?} = 0 AND '{?tagname?}'='')
OR (DocumentID IN (SELECT DocumentID FROM CMS_DocumentTag WHERE TagID = {?tagid|
(toint)?}))
OR (DocumentID IN (SELECT DocumentID FROM CMS_DocumentTag WHERE TagID IN (SELECT
TagID FROM CMS_Tag WHERE TagName = '{?tagname?}' AND TagGroupID = {?groupid|
(toint)?})))
```

The first line ensures displaying of all documents when **no query string parameters** (tagid nor tagname nor taggroupid) are received. The second line ensures displaying of documents based on the received **tagid** (or differently named parameter) from a **Tag cloud** web part. The third line ensures displaying of documents based on the received **tagname** and **groupid** from a **blog post's** Filed under section.

If you would like to learn more on the use of the **Tag cloud** web part, please refer to Community Site Guide -> Part 2 -> Pre-development tasks -> Creating the tag groups. This is a step-by-step tutorial on how to create a tag group.

If you would like to see examples of how the **Tag cloud** web part is added to the site, please refer to:

- Creating the Blogs section -> Creating the Blogs page in the same section of Community Site Guide
- Creating the Blog posts page
- Creating the News page

You will need to have the Community Site sample website installed to follow Kentico CMS Community Site Guide.

8.38 Time zones

8.38.1 Overview

The Time zones module enables the configuration of time zones for the physical location of the server, for particular websites and even for particular users. This can be useful if your site has an international audience and you want the date and time displayed on your site to be correct for users from across the world.

Before it can be used, the module must be enabled as described in the Enabling the module topic.

To learn how to configure individual time zones in the system, please see the Managing time zones topic. Time zones can also be set to adjust their time according to Daylight saving time.

Time zones are currently supported in:

- **CMS Desk -> Content**
- **CMS Desk -> My desk**
- **CMS Desk -> Tools ->** Booking system
- **Web parts** of the Blogs, Forums**,** Message boards**,** Messaging and Smart search modules; more information can be found in the Use in web parts topic

The Time zones internals and API sub-chapter provides information about the database tables and classes used by the module and examples of how time zones can be managed using the API and how correct time can be displayed by your code.

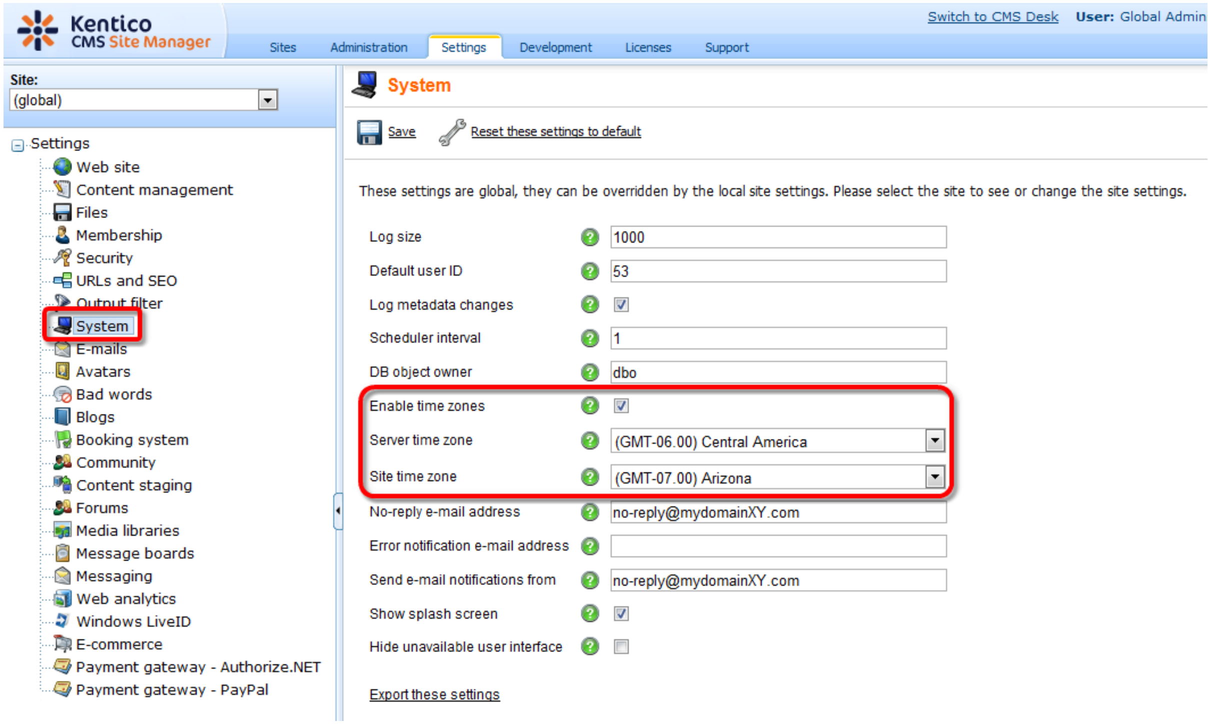
A typical example of use is displaying the time of forum posts when you have a global community – while the server may be located in New York (GMT -5:00), visitors coming from Paris (GMT +1:00) may see their new posts were added at 8am, while they would expect to see 2pm according to their current time.

Another example is a website of a global company that runs on a server in New York, but contains content for a French office. In this case, French visitors may wonder why the current time displayed by the server is 8am while it's 2pm in Paris. That's when you use the built-in support for multiple time zones.

8.38.2 Enabling the module

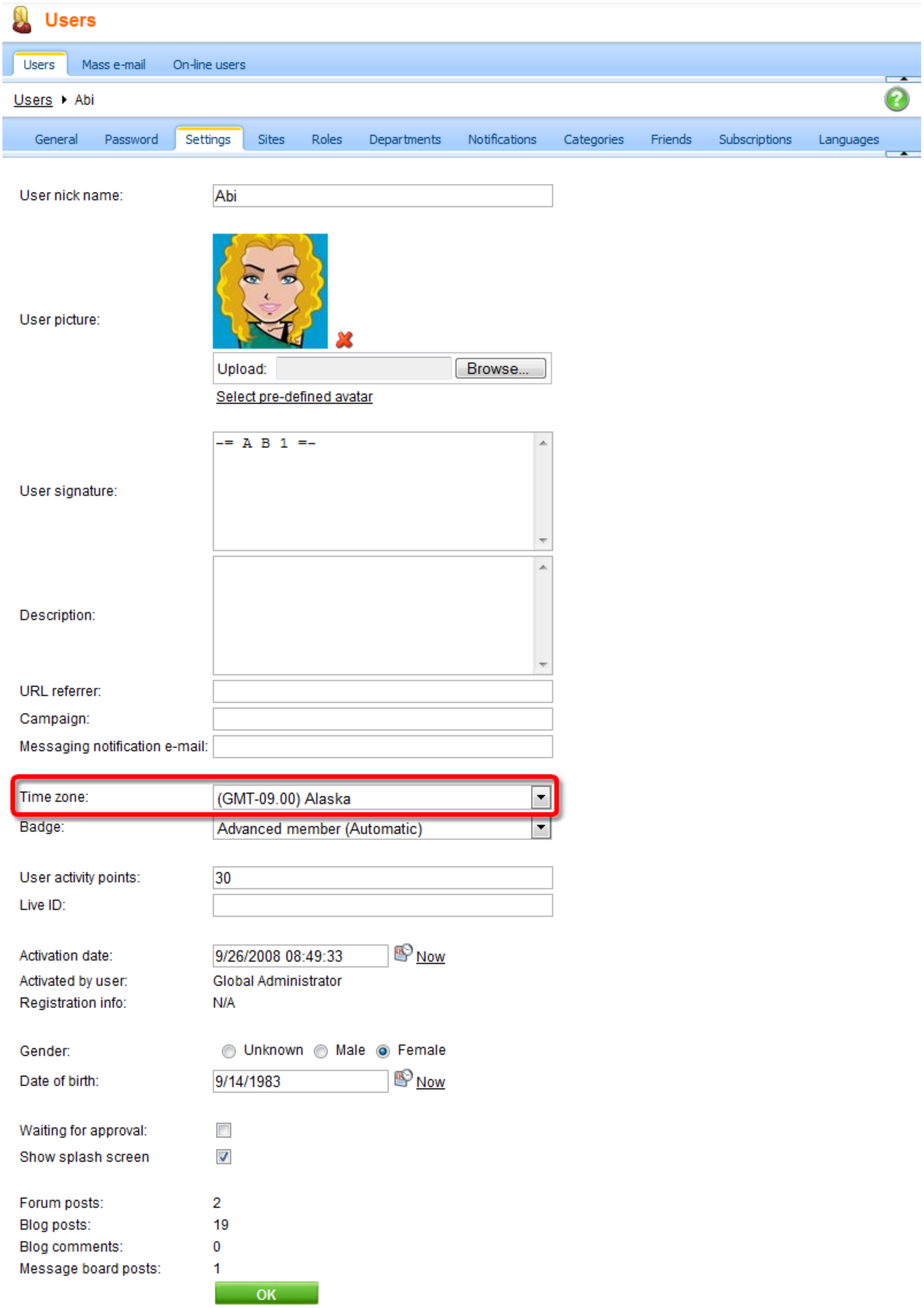
To enable the Time zones module, go to **Site Manager -> Settings -> System** and check the **Enable time zones** check-box. Below it, you can find the following two drop-down lists:

- **Server time zone -** time zone of the physical server location
- **Site time zone -** default time zone of the website selected by the **Site** drop-down list in the top-left part of the page; this time zone can also be set globally and inherited by sites that don't have it set differently



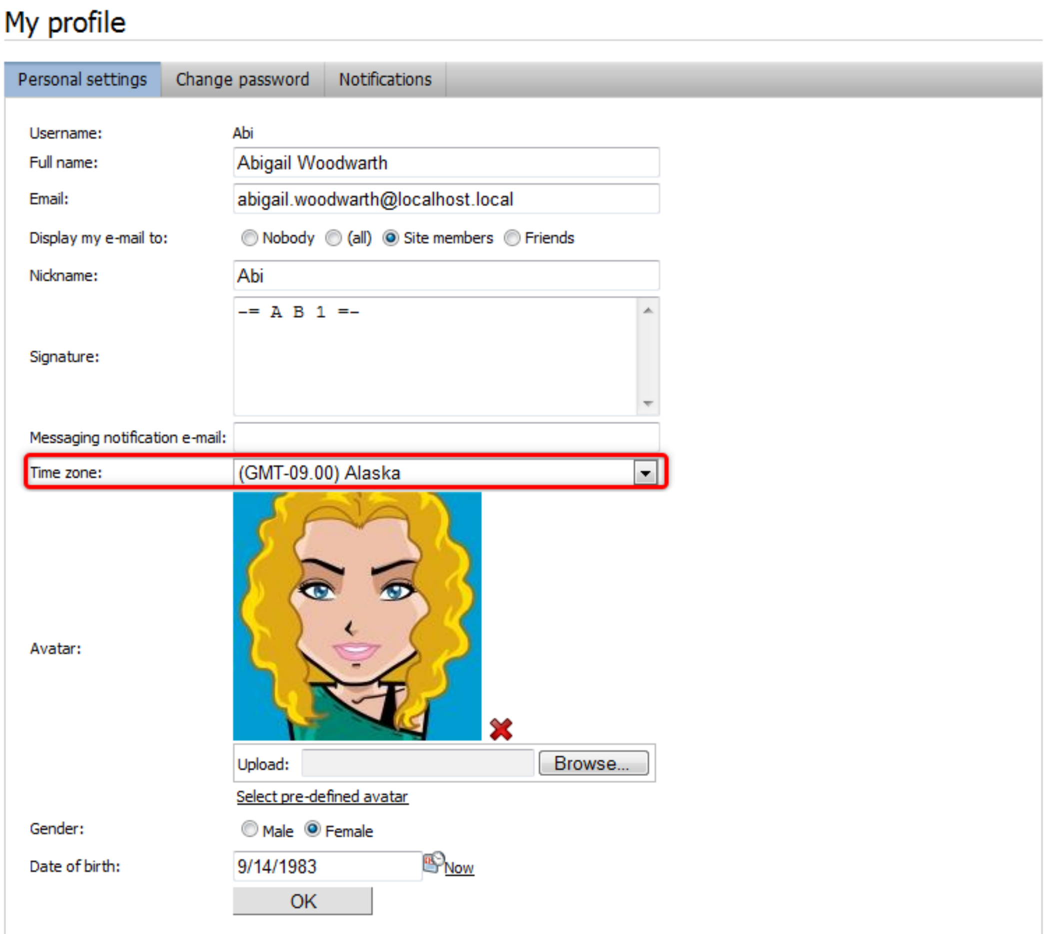
8.38.3 Setting user's time zone

Each user can have their own time zone settings. Where applicable, these time zone settings are used instead of the website's default time zone. A user's time zone can be set in both **CMS Desk** and **Site Manager**, on the **Administration -> Users -> Edit user -> Settings** tab. On the tab, the selection can be done using the **Time zone** drop-down list.



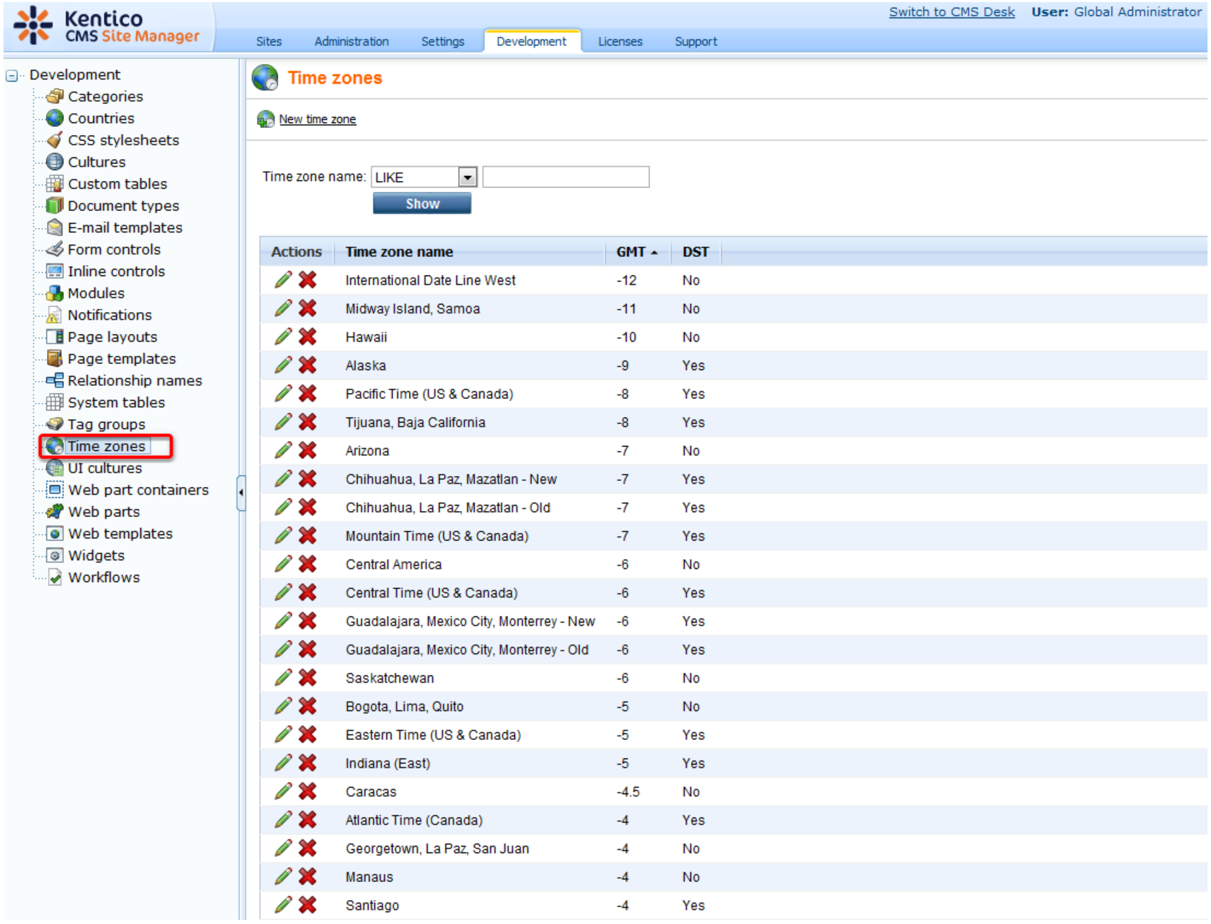
Users can also select their time zone on the live site if you place the **My Account** or **My Profile** web part on one of your pages. If they have access to **CMS Desk**, they can do this in **My Desk -> My profile**

-> Details. This is done the same way as above, by using the **Time zone** drop-down list.



8.38.4 Managing time zones

Time zones can be managed in **Site Manager -> Development -> Time zones**.



On this page, you can see a list of defined time zones. All time zones are displayed by default, but you can filter displayed items using the filter above the list. The only possible filtering parameter is the time zone **Display name**. If you type in the searched value and click the **Show** button, only those items that match the entered expression will be displayed in the list.

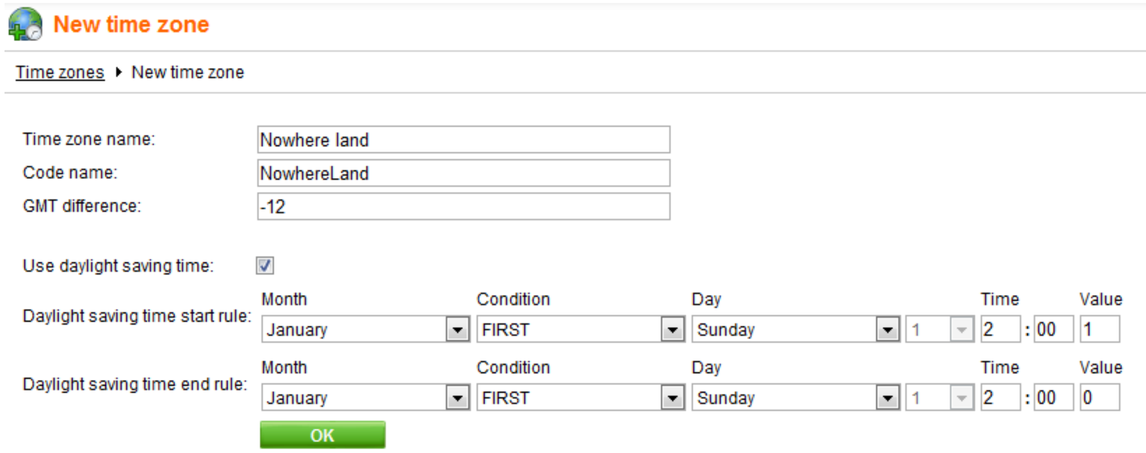
Time zones in the list can be **Edited** ( ) or **Deleted** ( ). You can also create new time zones by clicking the **New time zone** link.

Creating a new time zone

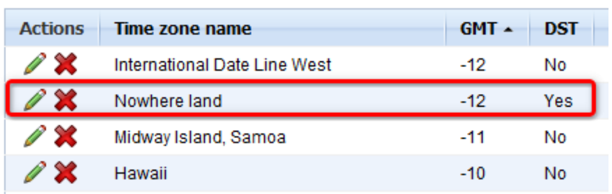
In the following example, you will learn how to create a new time zone:

1. Go to **Site Manager -> Development -> Time zones** and click the **New time zone** link.

2. Fill in the details that you can see in the following screenshot:



3. Click **OK**. You have just created the time zone. Now if you switch back to the time zones list, you should see the new time zone present among the records.



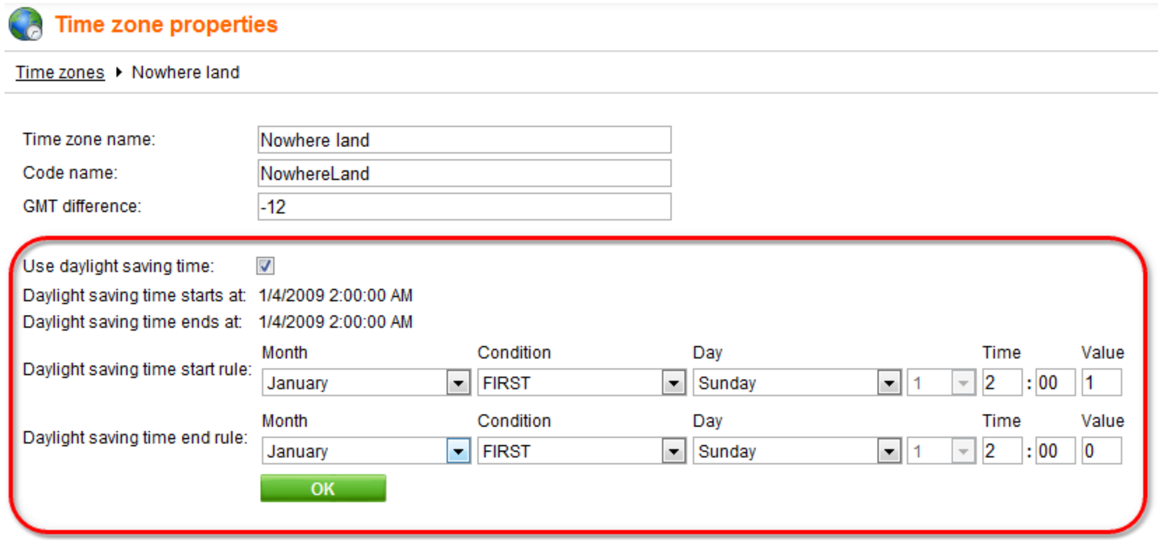
Actions	Time zone name	GMT	DST
	International Date Line West	-12	No
	Nowhere land	-12	Yes
	Midway Island, Samoa	-11	No
	Hawaii	-10	No

8.38.5 Daylight saving time

When creating a time zone or modifying some of the existing ones, you may come across the need to specify the daylight saving time (DST). This is a convention of setting clocks so that afternoons have more daylight and mornings have less of it. The amount of time advance and dates of change vary from country to country, however, it is usually a one hour advance at the beginning of spring and the advance is rolled back in autumn.

For more information about DST, please read this Wikipedia article: http://en.wikipedia.org/wiki/Daylight_saving_time

Daylight saving time can be set separately for each of the time zones. It can be set when creating a new time zone or when editing an existing one.



1. The first thing you need to do is to check the **Use daylight saving time** check-box. This enables DST for the time zone.

2. Now when you have DST enabled, you have to set the **DST start rule** and **DST end rule** for the current time zone. First, select the month in which the change will be carried out using the **Month** drop-down list.

3. Here comes the complicated part. You have to specify on which day of the selected month the change will be carried out. This is done by the **Condition** drop-down list and the two **Day** drop-down lists.

The following table explains the meanings of possible options for the **Condition** parameter:

FIRST	Day of the week can be selected. If you select Monday, the time advance will occur on the first Monday of the selected month.
LAST	Day of the week can be selected. If you select Monday, the time advance will occur on the last Monday of the selected month.
>=	Day of the week and day number can be selected. If you select Monday and 15, the time advance will occur on the first Monday after the 15th day of the selected month.
<=	Day of the week and day number can be selected. If you select Monday and 15, the time advance will occur on the last Monday before the 15th day of the selected month.
=	Day number can be selected. If you select 15, the time advance will occur on the 15th of the selected month.

4. Set the time when the change will occur on the specified date using the **Time** fields.

5. The last thing is to set the time difference between the standard time and DST. It should be entered into the **Value** field and represents the **difference from standard time in hours**. Use this value for the **DST start rule** and '**0**' for the **DST end rule**.

6. Click **OK** to save the settings.

8.38.6 Use in web parts

Web parts of the Blogs, Forums, Message boards, Messaging and Smart Search modules have the **Time zones** section in their **web part properties**, where the applied time zone can be set. The section contains the following two properties:

Time zone	• **Inherit** - inherits the time zone setting from the Page placeholder in which it is displayed (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	If you select one of the time zones, it will be used by the web part, regardless of user or website time zone settings.

In the case of the **Calendar** and **Event calendar** web parts, these web part properties take no effect. Instead, displaying of the correct time zone needs to be ensured in the used transformation, as described in the Displaying correct time in your code chapter.

8.38.7 Time zones internals and API

8.38.7.1 Database tables and API classes

The Time zones module uses the following database table:

- **CMS_TimeZone** - contains time zones and their settings

The Time zones API is provided by the following **CMS.SiteProvider** namespace classes:

- **TimeZoneInfo**, **TimeZoneInfoProvider** - these classes provide functionality for managing time zones
- **TimeZoneHelper** - this class contains further static methods for performing time zone related tasks

The following chapters show examples of how these classes can be used:

- Getting time zone data
- Managing time zones
- Converting time according to a time zone

The following chapter shows how correct time converted according to time zone settings can be displayed in transformations and by user controls:

- Displaying correct time in your code

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.38.7.2 Getting time zone data

The following sample code shows how you can get a time zone as a *TimeZoneInfo* object:

[C#]

```
using CMS.SiteProvider;

...

        // Get time zone object by time zone code name
        TimeZoneInfo tzi = TimeZoneInfoProvider.GetTimeZoneInfo("Hawaii");

        // Get time zone object by ID
        TimeZoneInfo tzi2 = TimeZoneInfoProvider.GetTimeZoneInfo(1);
```

The following sample code shows how you can get a DataSet containing time zones from the system:

[C#]

```
using System.Data;
using CMS.SiteProvider;

...

        string where = "TimeZoneID > 10";
        string orderBy = "TimeZoneDisplayName";
        int topN = 10;
        string columns = "TimeZoneID, TimeZoneDisplayName, TimeZoneGMT";

        // Get dataset of time zones according to the given parameters
        DataSet ds = TimeZoneInfoProvider.GetTimeZones();

        DataSet ds2 = TimeZoneInfoProvider.GetTimeZones(where, orderBy);

        DataSet ds3 = TimeZoneInfoProvider.GetTimeZones(where, orderBy, topN, column
```

8.38.7.3 Managing time zones

The following sample code shows how you can create a new time zone and add it to the system:

[C#]

```
using CMS.SiteProvider;

...

        // Create new object
        TimeZoneInfo tzi = new TimeZoneInfo();

        // Set properties
        tzi.TimeZoneName = "TestTimeZone";
        tzi.TimeZoneDisplayName = "Test time zone";
        tzi.TimeZoneGMT = 2;
        tzi.TimeZoneDaylight = true;
        tzi.TimeZoneRuleStartRule = "MAR|SUN|1|LAST|3|0|1";
        tzi.TimeZoneRuleEndRule = "OCT|SUN|1|LAST|3|0|0";
        tzi.TimeZoneRuleStartIn = TimeZoneInfoProvider.CreateRuleDateTime(tzi.TimeZo
        tzi.TimeZoneRuleEndIn = TimeZoneInfoProvider.CreateRuleDateTime(tzi.TimeZone

        // Save to database
        TimeZoneInfoProvider.SetTimeZoneInfo(tzi);
```

The following sample code shows how you can modify the *Display name* of an existing time zone:

[C#]

```
using CMS.SiteProvider;

...

        // Get time zone object
        TimeZoneInfo tzi = TimeZoneInfoProvider.GetTimeZoneInfo("TestTimeZone");

        // If time zone exists
        if (tzi != null)
        {
            // Update property
            tzi.TimeZoneDisplayName += " updated";

            // Save to database
            TimeZoneInfoProvider.SetTimeZoneInfo(tzi);

        }
```

The following sample code shows how you can delete an existing time zone:

[C#]

```
using CMS.SiteProvider;

...
```

```
        // Get time zone object
        TimeZoneInfo tzi = TimeZoneInfoProvider.GetTimeZoneInfo("TestTimeZone");

        // If time zone exists
        if (tzi != null)
        {
            // Delete from database
            TimeZoneInfoProvider.DeleteTimeZoneInfo(tzi);
        }
```

8.38.7.4 Converting time according to a time zone

The following sample code shows how to get the current time according to a user's time zone settings:

[C#]

```
using CMS.SiteProvider;

...

        // Get user
        UserInfo ui = UserInfoProvider.GetFullUserInfo("administrator");

        // If user exist
        if (ui != null)
        {
            // Get converted time
            System.DateTime convertedTime = TimeZoneHelper.ConvertUserDateTime(Syste
        }
```

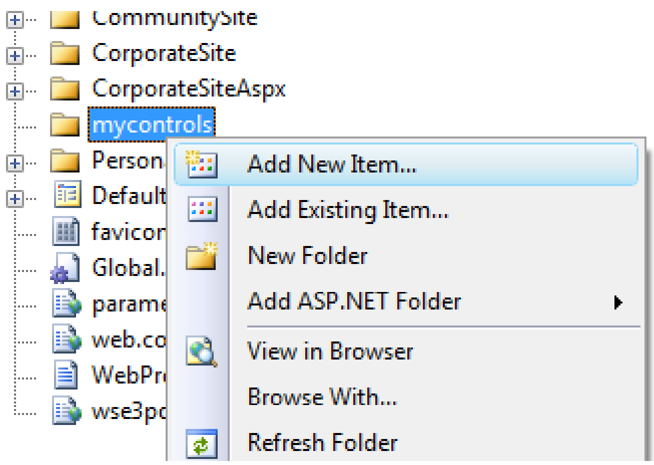
8.38.7.5 Displaying correct time in your code

The following methods can be used in transformation code to display the correct time according to time zone settings.

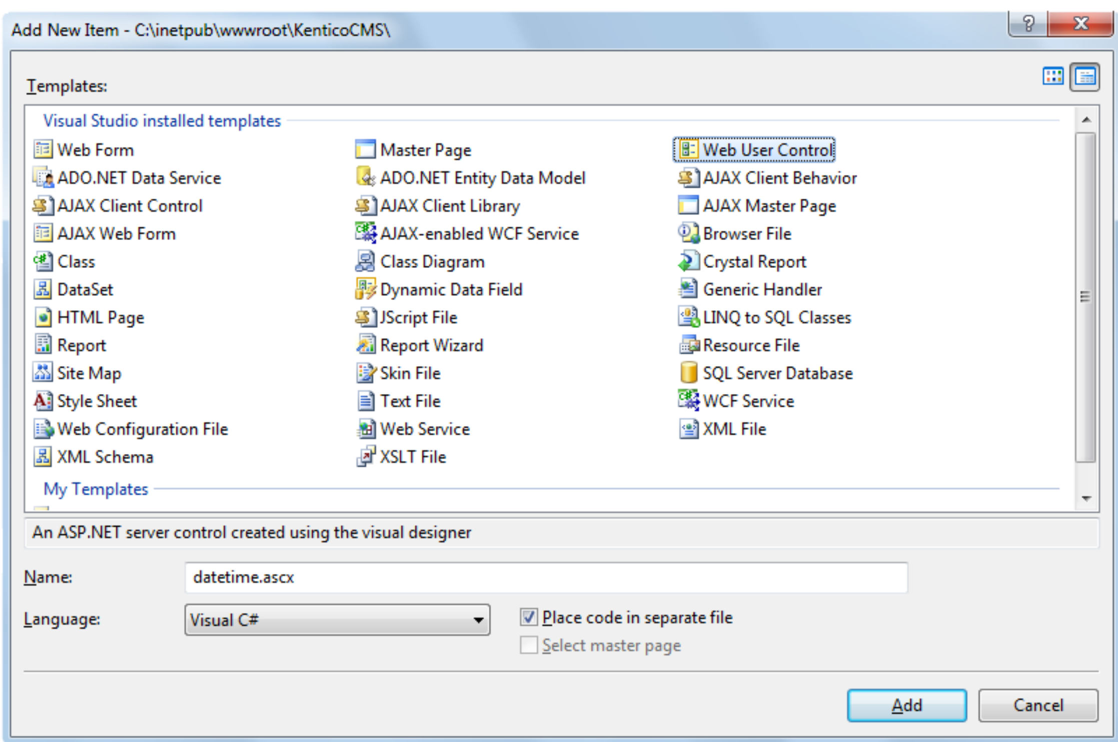
<%# GetDateTime (DateTime.Now) %>	Returns date-time value according to web part time zone settings.
<%# GetSiteDateTime (DateTime.Now) %>	Returns date-time according to site time zone settings.
<%# GetUserDateTime (DateTime.Now) %>	Returns date-time according to the current user's time zone settings.
<%# GetCustomDateTime (DateTime.Now, "GreenwichMeanTime") %>	Returns date-time according to the time zone given in the second parameter.

In the following example, you will learn how to use the **General -> User control** web part to display the current date and time on your site, which will reflect the time zone settings of the web part.

1. Open the web project in Visual Studio and create a new subfolder in the project folder. Name it *mycontrols*. Right click the folder and click Add new item.



2. Create a new WebUserControl in the mycontrols folder, name it *datetime.ascx*.



3. Edit the user control on the Design tab. Drag and drop a **Label** control onto the form.

4. Add the following code to the **PageLoad** method of the user control:

[C#]

```
Label1.Text = CMS.CMSHelper.CMSContext.ConvertDateTime(DateTime.Now, this).
ToString();
```

5. Add the **General -> User control** web part somewhere to some page of your website. Set the following properties of the web part:

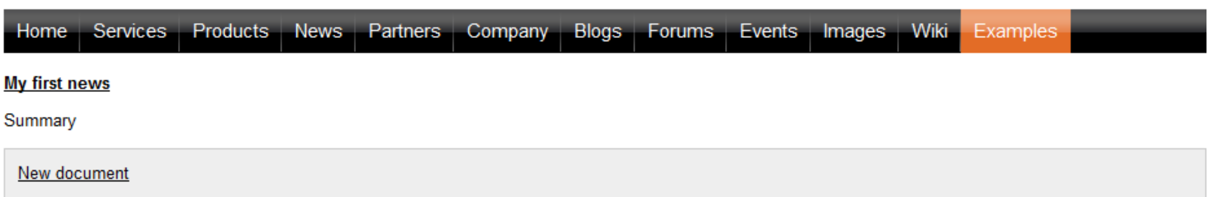
- **User control virtual path:** *~/mycontrols/datetime.ascx*
- **Time zone**: *Custom*
- **Custom time zone**: any time zone of your choice

and click **OK**. If you switch to the live site now, you should see the web part displaying the current date and time in the selected time zone. Now you can also try changing the value of the **Custom time zone** property and verify that the time displayed on the live site changes according to it.

8.39 User contributions (Wiki)

8.39.1 Overview

The User contributions module allows you to create content editing interface for site members. It means that chosen website visitors can create, edit and delete content, even if they are not editors and cannot access CMS Desk.



There are several scenarios where you can use this module, for example:

1. Community news

You can create a list of news and allow community members to add news without going to CMS Desk.

2. Partner directory

Your business partners can manage their profile on your website and the list of their reference project.

3. Business directory

You can create a business directory for some town or industry and let the business owners to manage their own profile.

Implementation

You can use the following web parts to implement user contributions:

- **Contribution list** - this web part allows you to display a list of contributions (documents) and the **New document** link.

- **Edit contribution -** this web part allows you to edit an existing document.

You can find the description of both web parts in the Kentico CMS Web Parts reference.

- If you would like to see an example of how to create a new section for publishing community news, please refer to the Example: Publishing community news topic.
- If you would like to see an example of how to create a list of partners who will be able to edit their profile on a special page after they have signed in, please refer to the Example: Editing partner profile topic.
- If you would like to learn about the security possibilities of the User contributions (Wiki) module, please refer to the Security topic.
- A brief reference on the relationship between the User contributions (Wiki) module and API can be found in the User contributions and API topic.

Several more practical examples of the use of the User contributions (Wiki) module are available in Kentico CMS Community Site Guide; please note that these examples do not concern the whole functionality of the module but focus on its use in a broader context of the Community Site sample website:

- See Community Site Guide -> Part 2 -> Creating the Blogs section -> Creating the Blogs page: An example of the use of the User contributions (Wiki) module web parts in the context of creating the Blogs section.
- See Creating the Groups section -> Preparing the Group pages section -> Creating the Pages page in the same section of Kentico CMS Community Site Guide: An example of the use of the User contributions (Wiki) module web parts in the context of creating the Groups section.

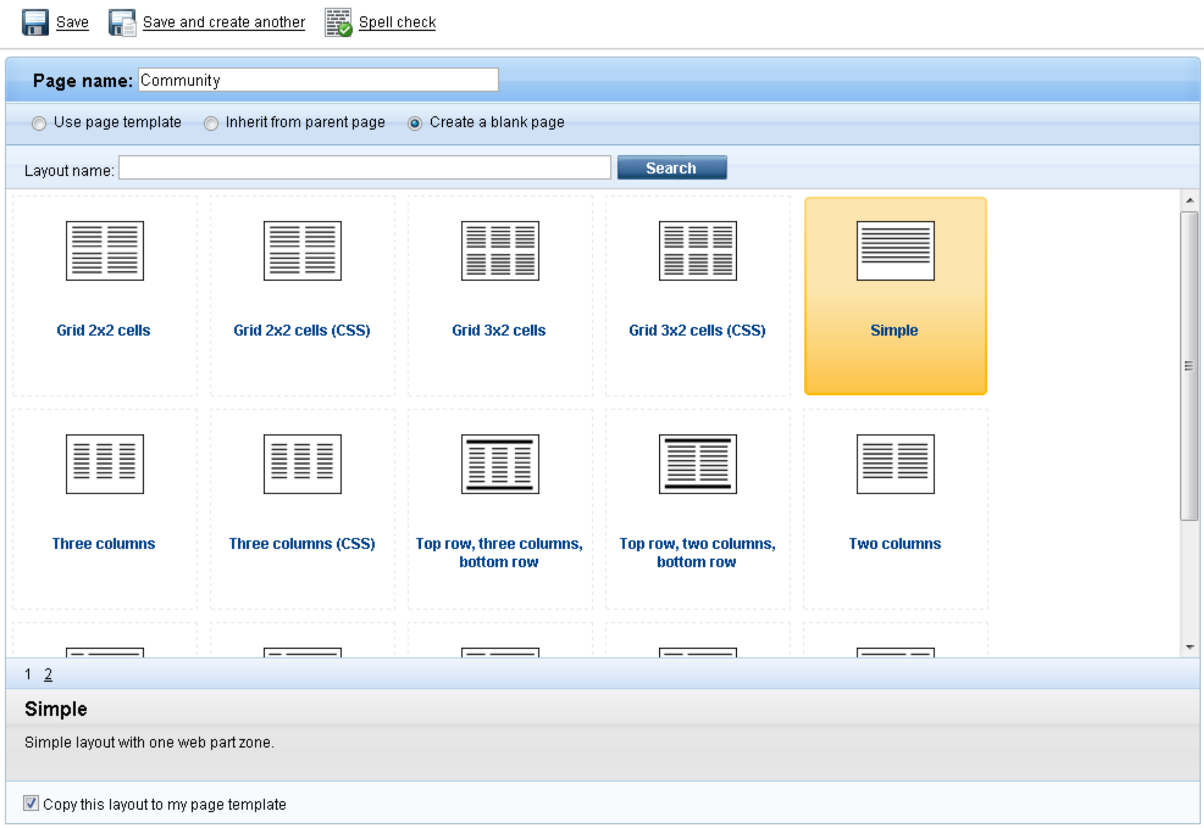
8.39.2 Example: Publishing community news

In this example, you will create a new section for publishing community news. Each registered site member can create new news items and edit/delete their previously posted news. The example assumes that you are using the Corporate Site sample website. In this topic you will learn how to:

- Create the community news section
- Add the New document button
- Add the editing support
- Create a testing user
- Create your first news item
- Approve the news

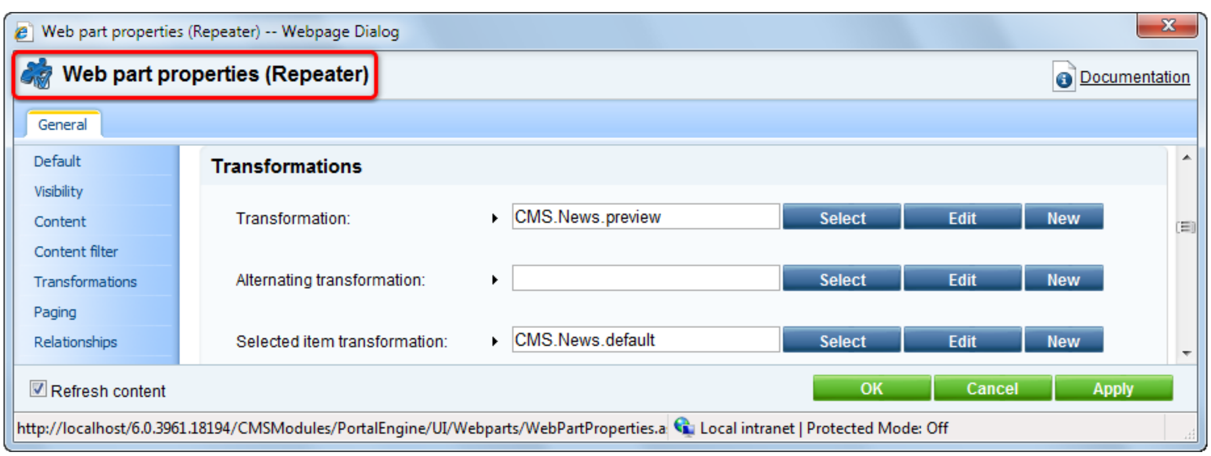
Creating the community news section

1. Sign in as administrator to **CMS Desk**, go to the **Content** section and click **Examples** in the content tree. Click **New** and choose to create a new **Page (menu item)**. Enter the name **Community** and choose to create a blank page using the **Simple** layout:



2. Click **Save**. Switch to the **Design** tab of the newly created page and add the **Listings and Viewers/Repeater** web part. Set the following properties:

- **Web part control ID:** repeaterNews
- **Document types:** cms.news
- **Transformation:** CMS.News.preview
- **Selected item transformation:** CMS.News.default



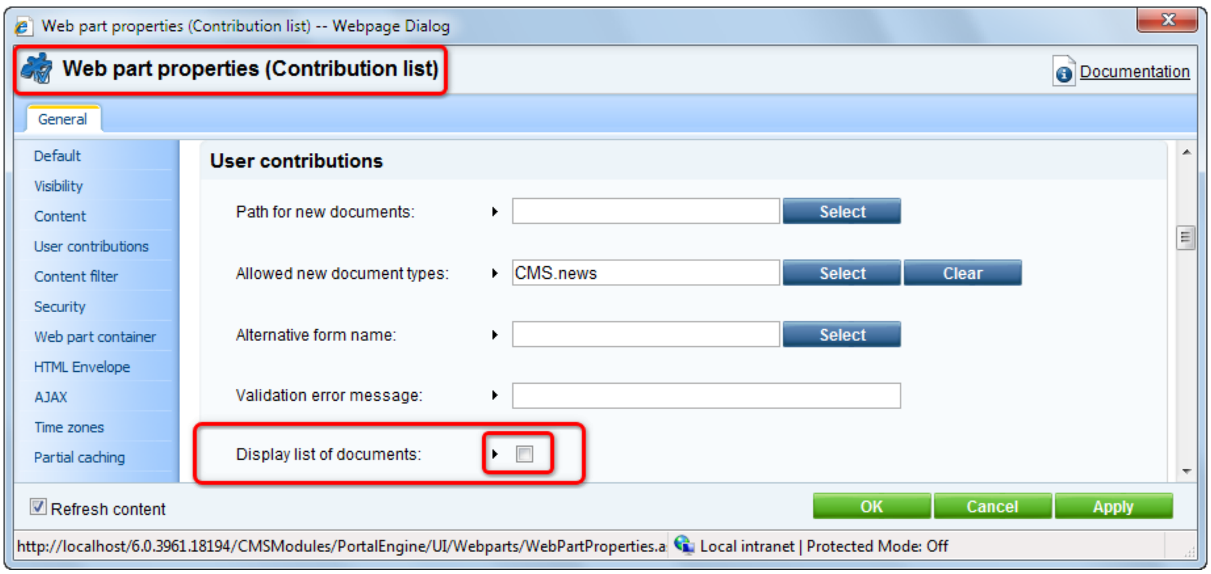
3. Click **OK**.

Adding the New document button

1. Add a new web part **User contributions/Contribution list**. It will display the **New document** link. Set the following values:

- **Show for document types:** cms.menuitem
  - this ensures that the web part is displayed only when the list of news is displayed, not on the news detail page.
- **Allowed new document types:** CMS.News
  - this means that the users will be allowed to create only news items under this section.
- **Display list of documents:** no (unchecked)
  - this ensures that the web part displays only the ***New document*** *link, without displaying the list of documents.*
- **Allow editing by users:** Authenticated
  - this means that only authenticated users will be able to edit/delete the document. Authenticated users are those who are signed-in to the website (not to the administration interface).

2. Leave the other values as they are by default.



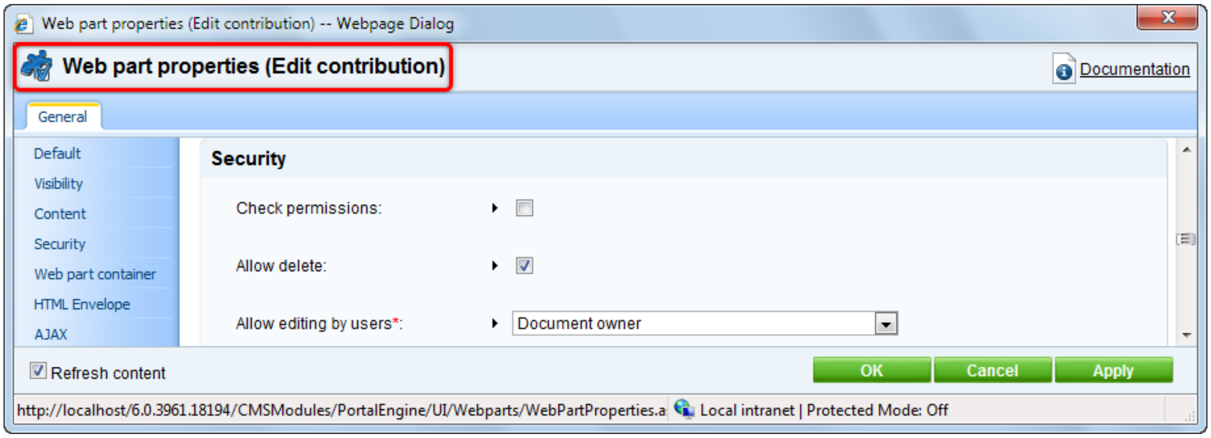
3. Click **OK**.

Adding the editing support

1. Add a new web part **User contributions/Edit contribution**. It will display the **Edit/Delete** icons when some news item is chosen.

- **Show for document types:** cms.news
  - this ensures that the web part is displayed only on the news detail page.
- **Allow editing by users:** Document owner

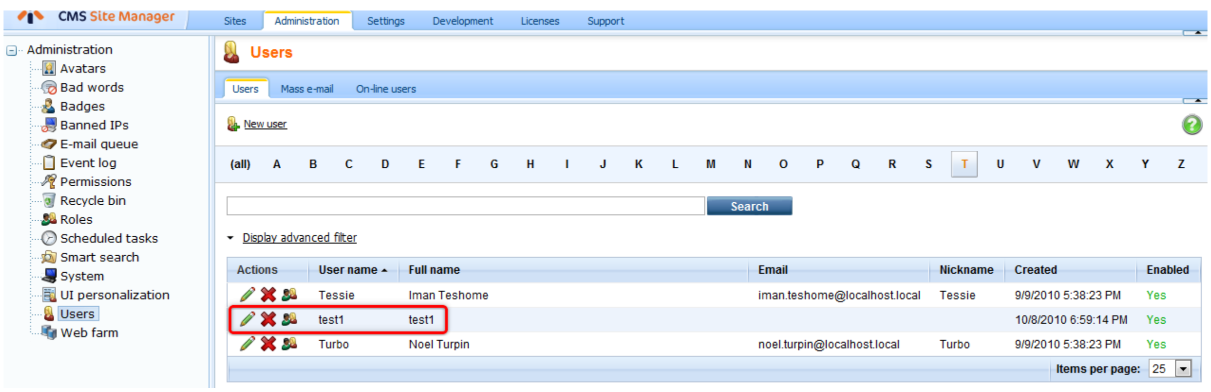
Please note that the **Edit contribution** web part is missing the **Allow insert/edit** properties.



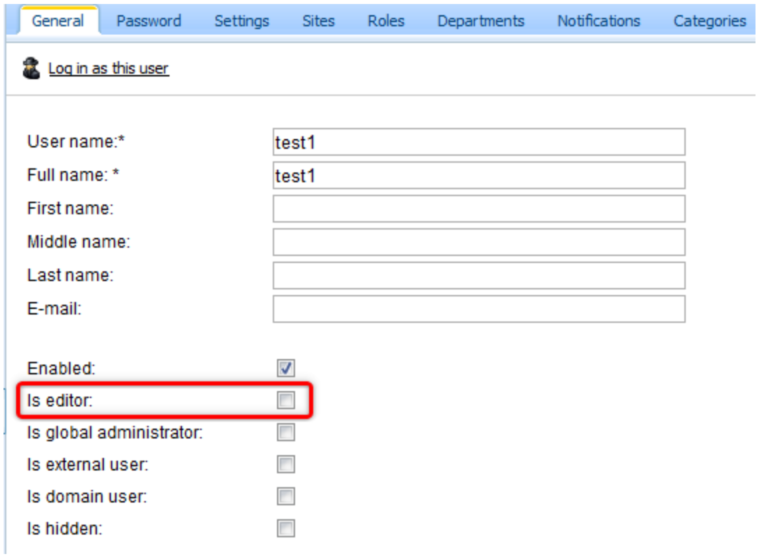
2. Click **OK**.

Creating a testing user

1. Go to **Site Manager -> Administration -> Users** and click the **New user** link to create a new user with user name **test1**. Please note that to create a user, at least the values of the following two compulsory fields: **User name** and **Full name** must be entered.



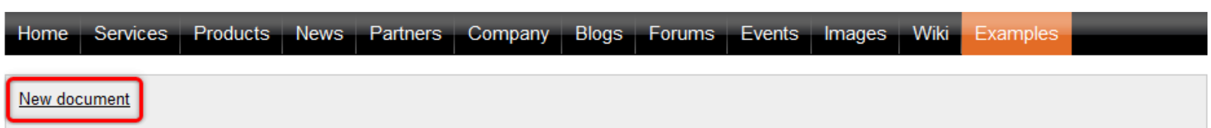
2. For the purpose of this example, choose no (uncheck the box) in the **Is editor** field - **test1** is a common site member without access to Kentico CMS Desk.



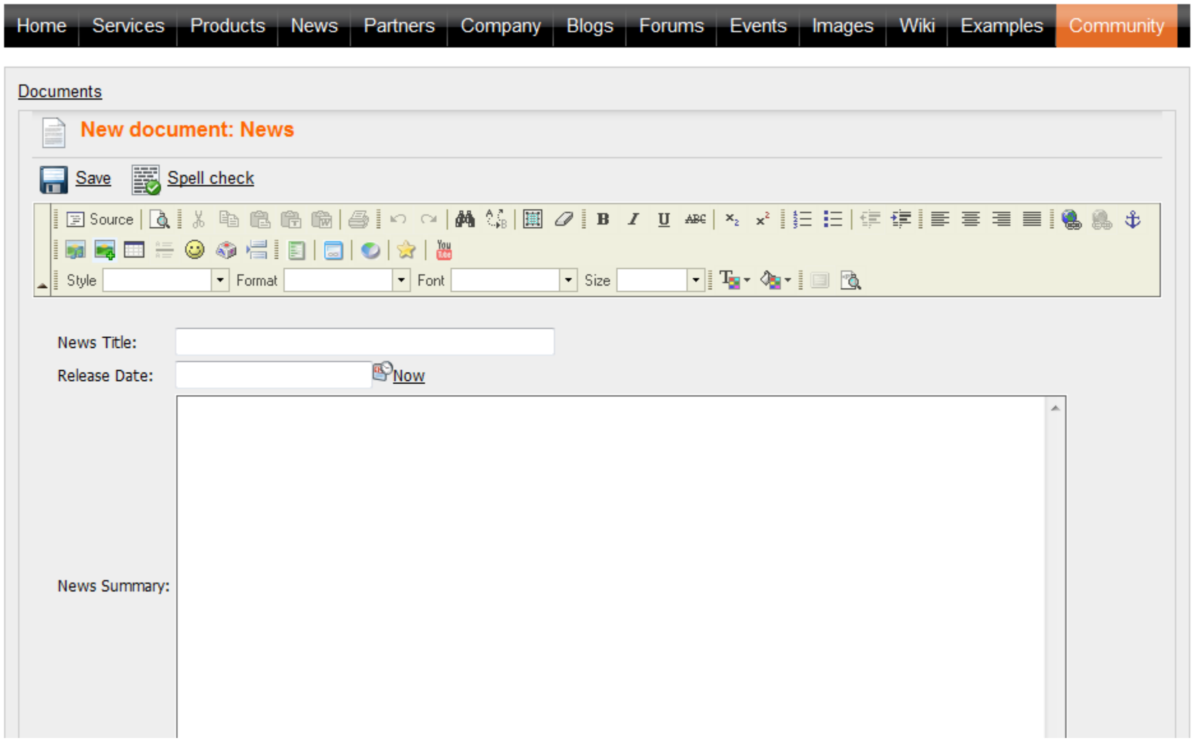
Click **OK** to save the changes.

Creating your first news item

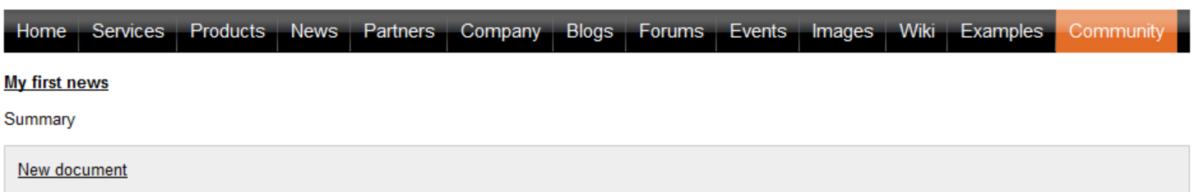
1. Sign out and go to the **Examples -> Community** section. You will see a blank page since the user contributions are now enabled only for site members. In order to sign in, click the **My account** link at the top and sign in as user **test1**. Then go back to the **Examples -> Community** section. You will see the **New document** link:



2. Click this link and you will be displayed with the news editing form:

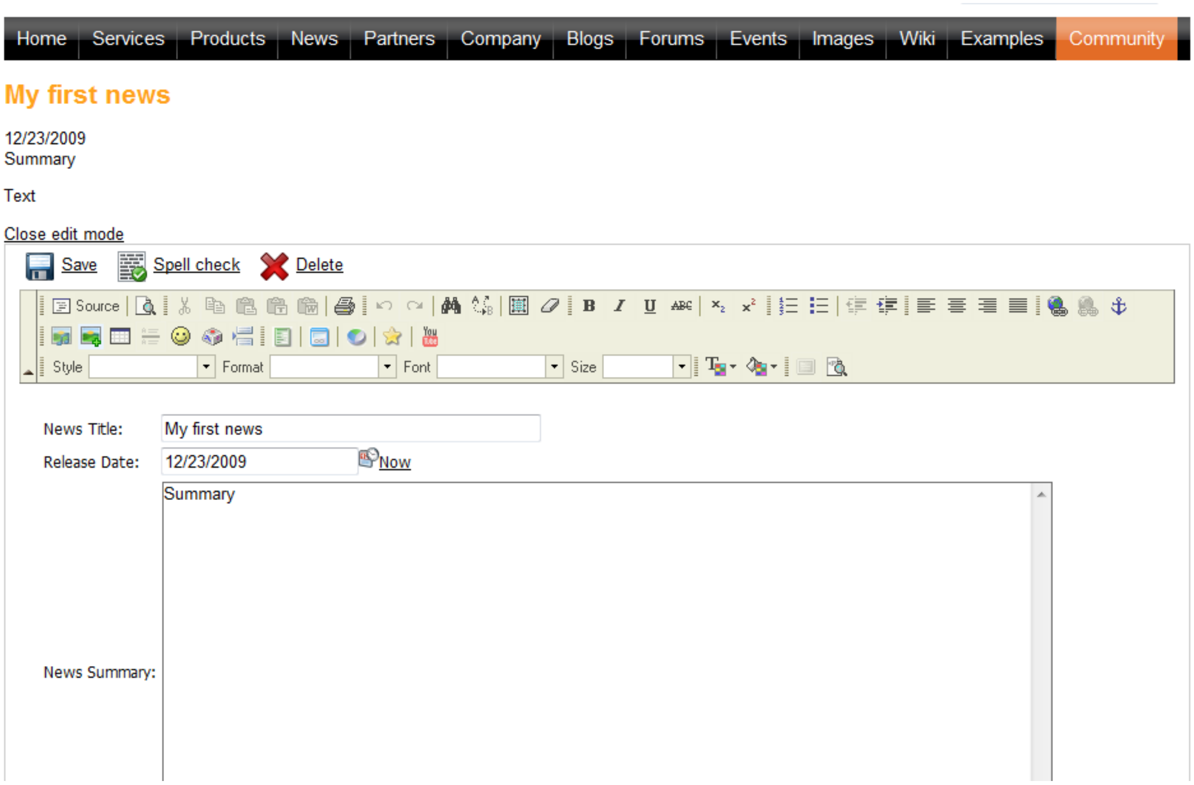


3. Enter some content and click **Save**. Now select **Examples -> Community** from the main menu at the top. You will see the **Community** page with your first news item:



4. Click the news item link to see the details page. The **Edit** and **Delete** links are be displayed below the news item:

5. Choose to **Edit** the news item and you will be displayed with the editing form where you can edit its content:



6. Make some modifications and click **Save**. Click the **Close edit mode** link to see the updated page.

Approval process

If you need to enforce some approval process for published news, you can simply set up workflow for the given site sections and all news will need to be approved by a site manager. You can find more details on workflow configuration in the Workflow overview chapter.

In this example, you have learned how to create a site section where community members can create and edit content without having access to Kentico CMS Desk.

8.39.3 Example: Editing partner profile

In this example, you will create a list of partners who will be able to edit their profile on the **My profile** page after they have signed in. This example assumes that you are using the Corporate Site sample website.

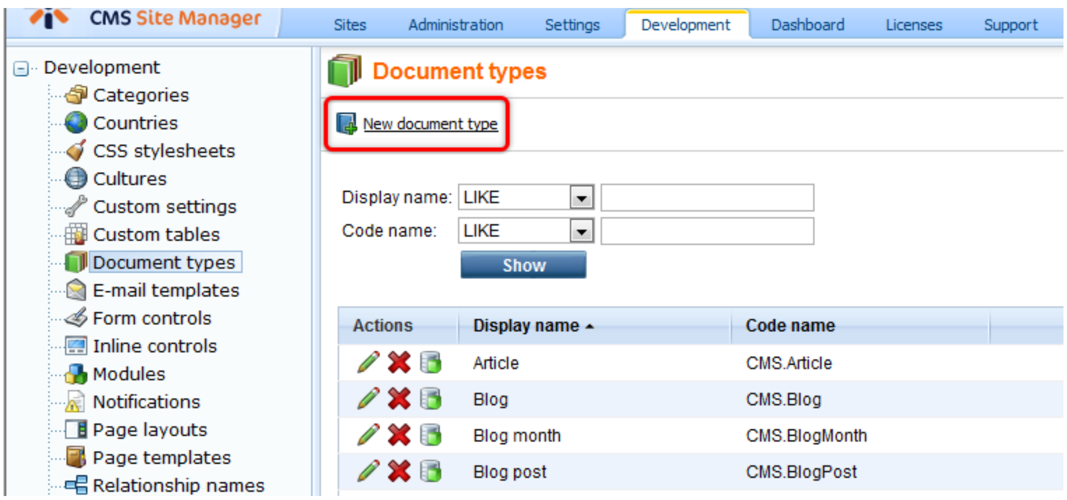
- Create the partner document type
- Create the partner list
- Create the partner user account
- Create the partner profile document
- Create the partner profile editing page
- Test the profile editing page

Creating the partner document type

Please note

A detailed description of how to create a new document type can be found in the Document types and transformations -> Defining a new document type topic in the Development section of the Developer's Guide.

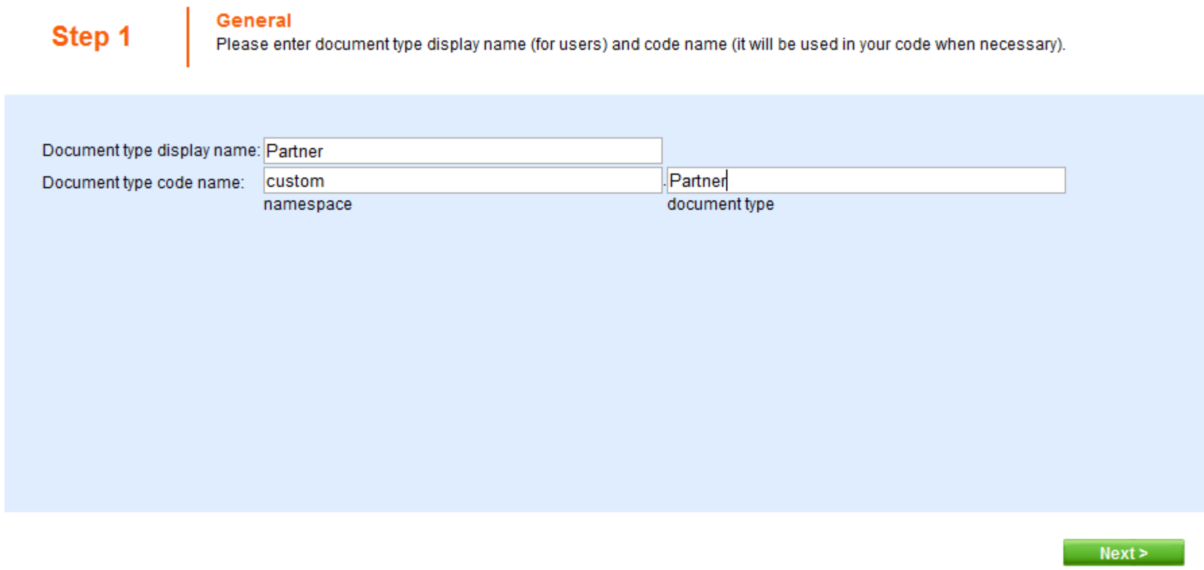
Before you create the page, it is necessary that you create a new document type **Partner**. Sign in to **Site Manager** and go to **Development -> Document types** where you need to click the **New document type** link.



Step 1: General

Enter the following values and click the **Next** button.

- **Document type display name:** Partner (this name will be displayed to the users)
- **Document type code name:** custom.Partner

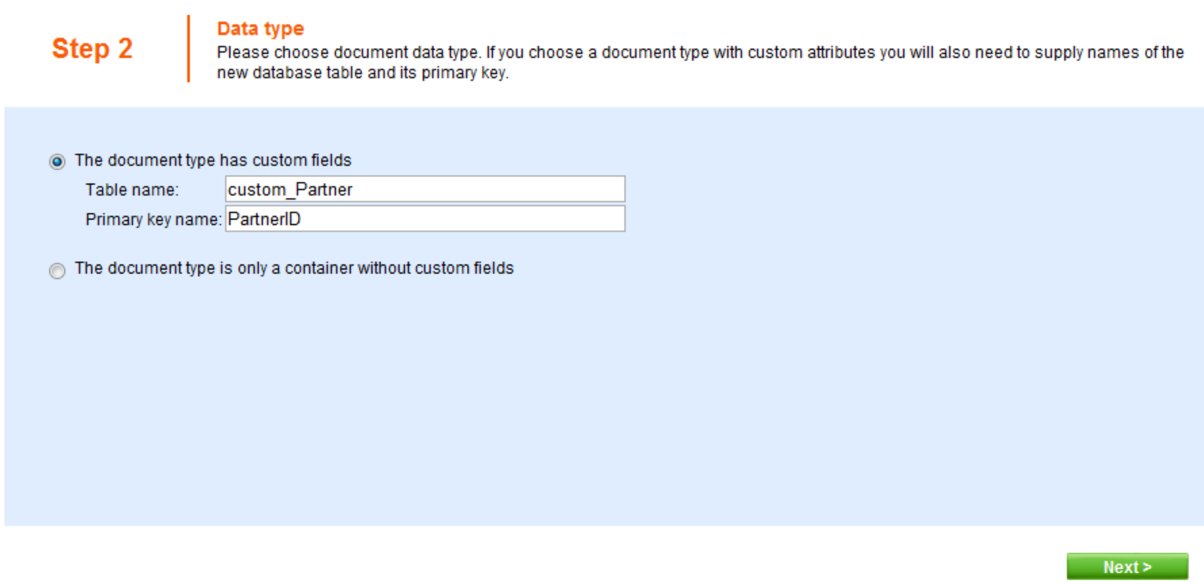


Please note that *custom* in the **Document type code name** field is your namespace to distinguish your document types from system types that use the *cms* namespace. You will use this value in web part properties later.

Step 2: Data type

You need to choose the name of the database table that will be used for storing partner details. You also need to enter the name of the primary key in this table. Choose the option **The document type has custom fields** and enter the following values:

- **Table name:** custom_Partner
- **Primary key name:** PartnerID



Click **Next**.

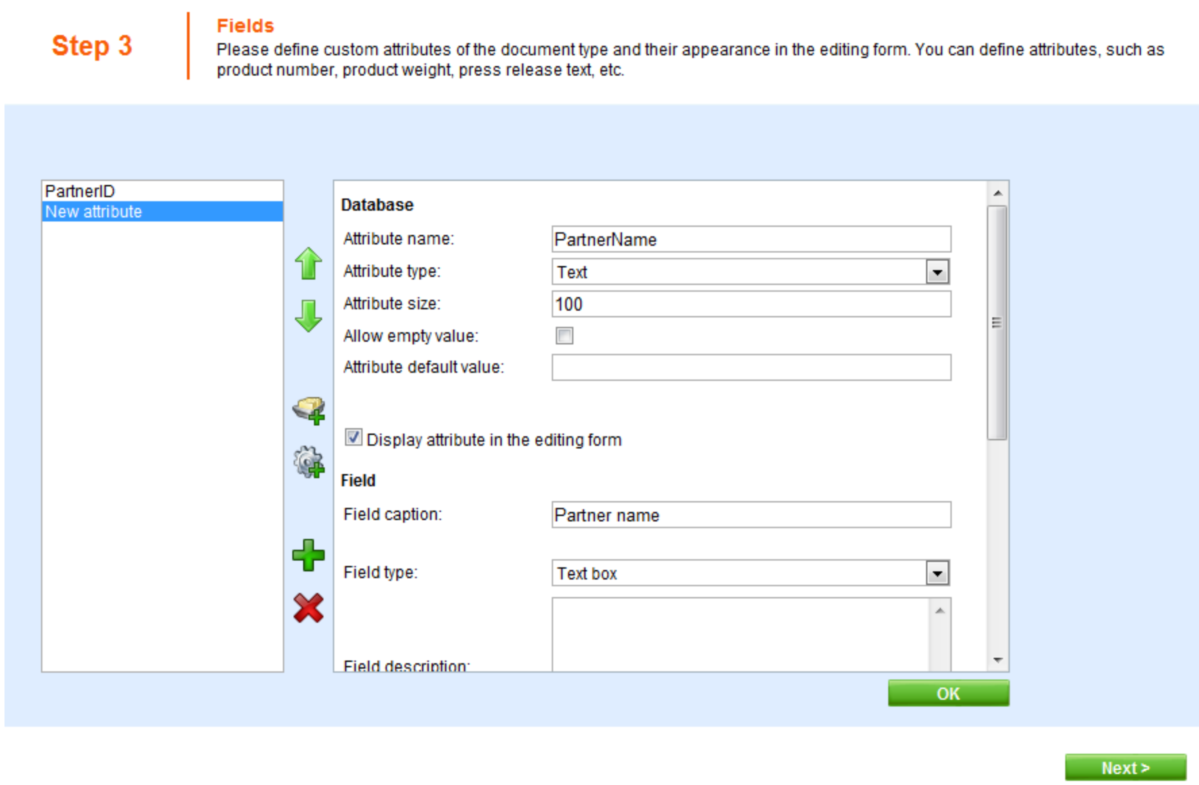
Step 3: Fields

Now you need to define the columns of the table or fields. Click **New attribute** () to create a new field, enter the following values and click **OK**.

- **Attribute name:** PartnerName
- **Attribute type:** Text
- **Attribute size:** 100
- **Field caption:** Partner name
- **Field type:** TextBox

- **Attribute name:** PartnerProfile
- **Attribute type:** Long Text
- **Field caption:** Partner profile
- **Field type:** HTML area (Formatted Text)

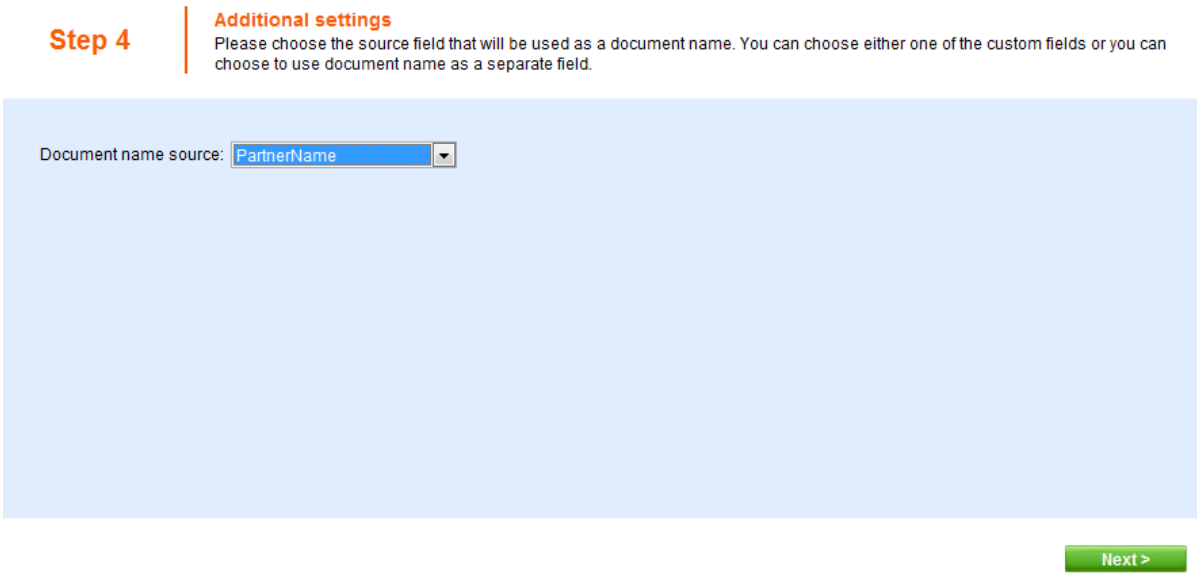
When you have defined both the fields, click **Next** again.



Step 4: Additional Settings

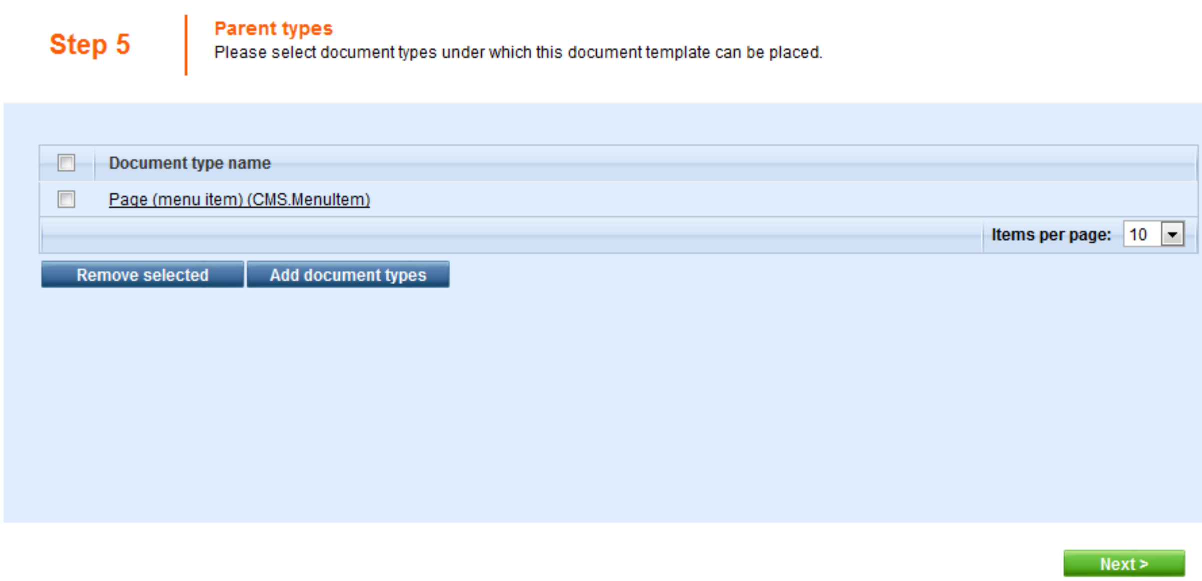
Now you need to choose the field that will be used as document name. Choose the *PartnerName* field option from the **Document name source** drop-down list and click **Next**.

- **Document name source:** PartnerName



Step 5: Parent types

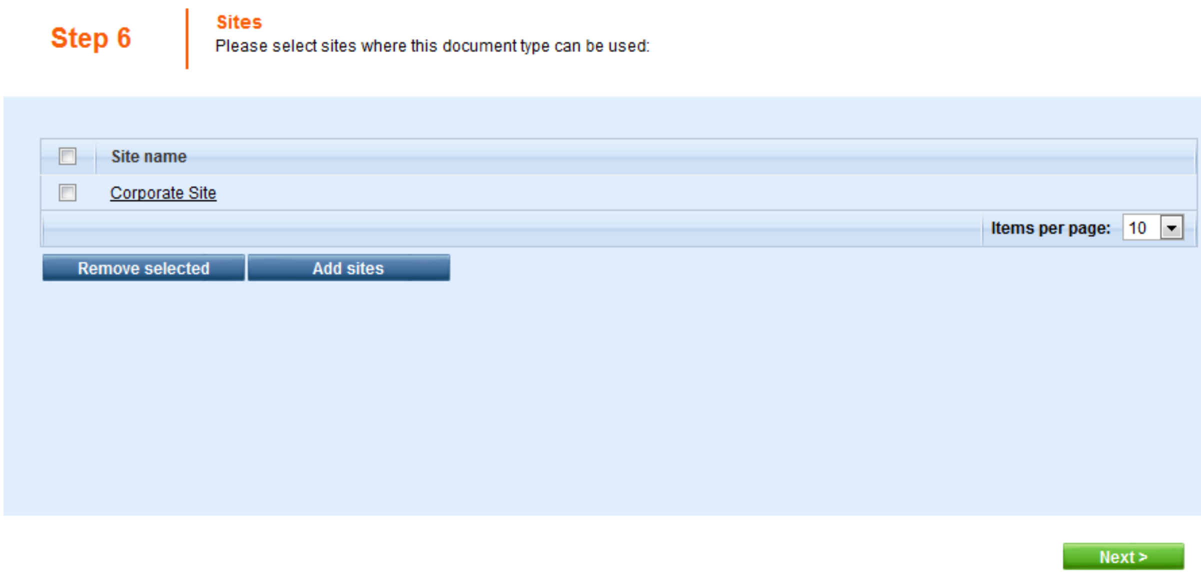
In this step, you need to select the document types under which the partner documents will be displayed. Check only the **Page (menu item)** value, which means the users will be able to create computer documents only under some page, not under article or news document in the content tree. Click the **Next**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



Step 6: Sites

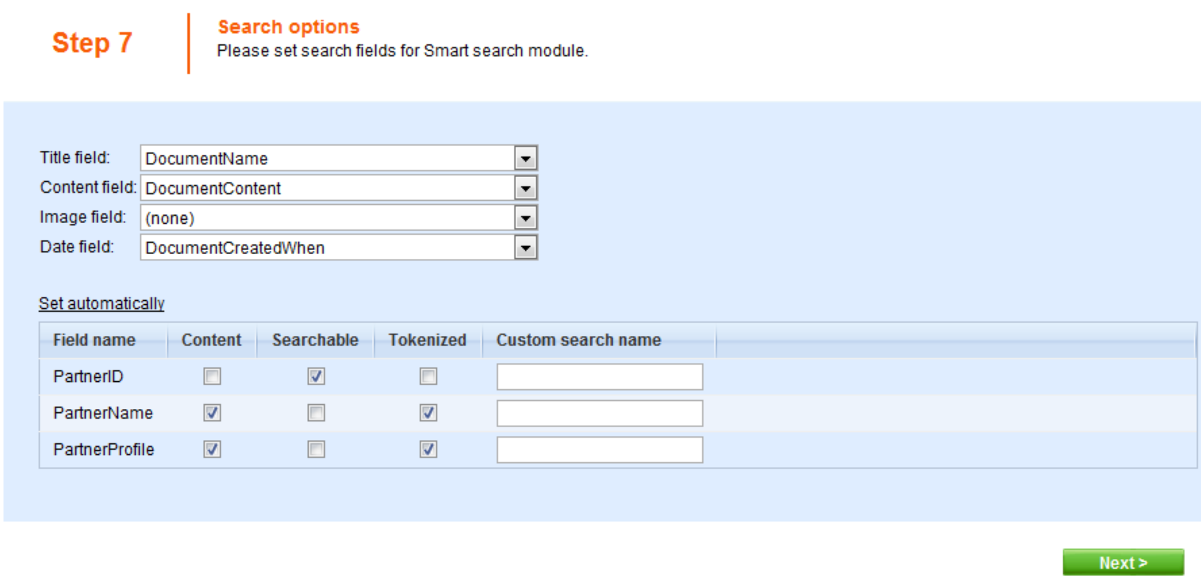
Here you need to choose which websites will use this document type. Choose your current website and

click **Next**.



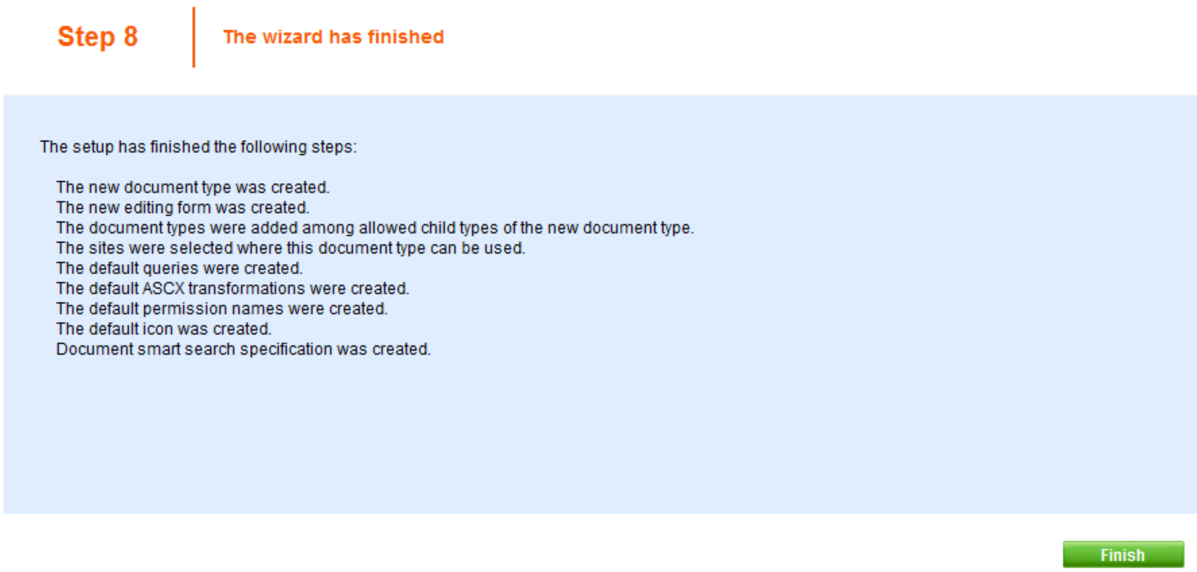
Step 7: Search options

In this step, you are asked to specify how documents of this type will be indexed and displayed in the search results. For more information on these settings, please refer to the Settings for particular object types topic. Make your choice and click **Next**.



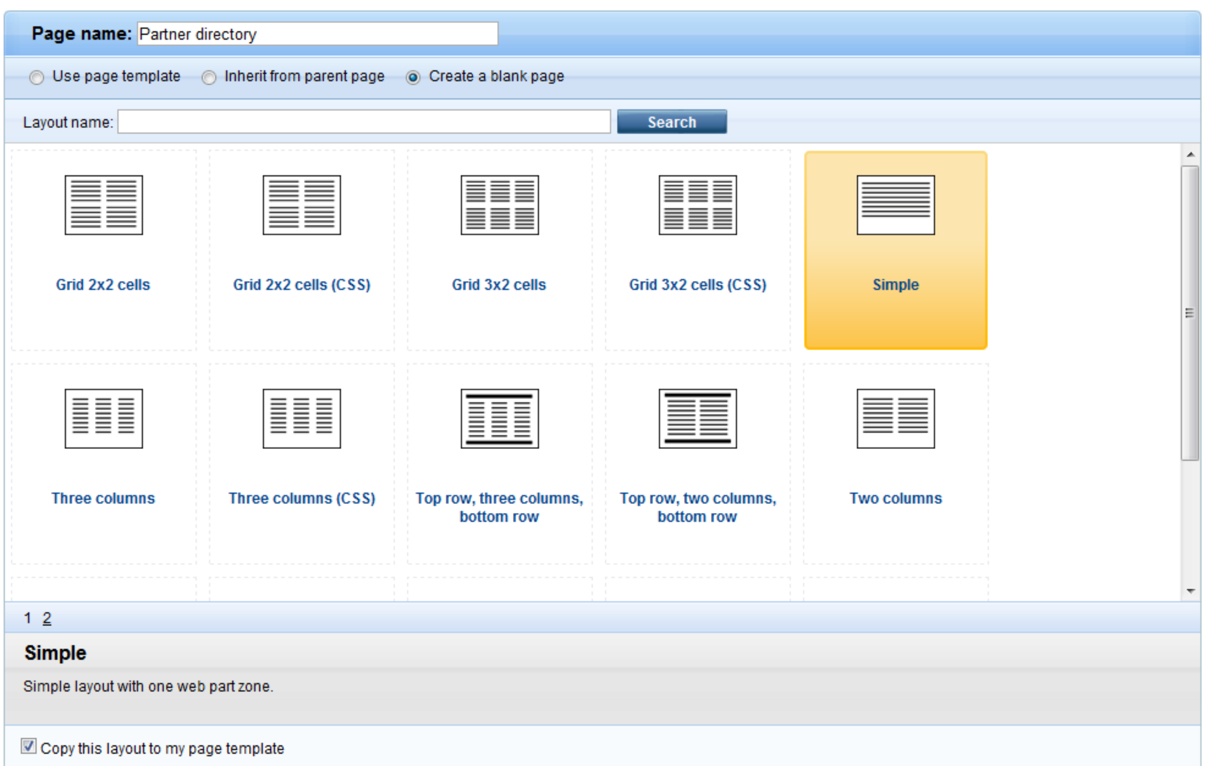
Step 8: The wizard has finished

Step 8 is the last step in the process of creating the partner document type - the wizard has finished the configuration of the this document type.



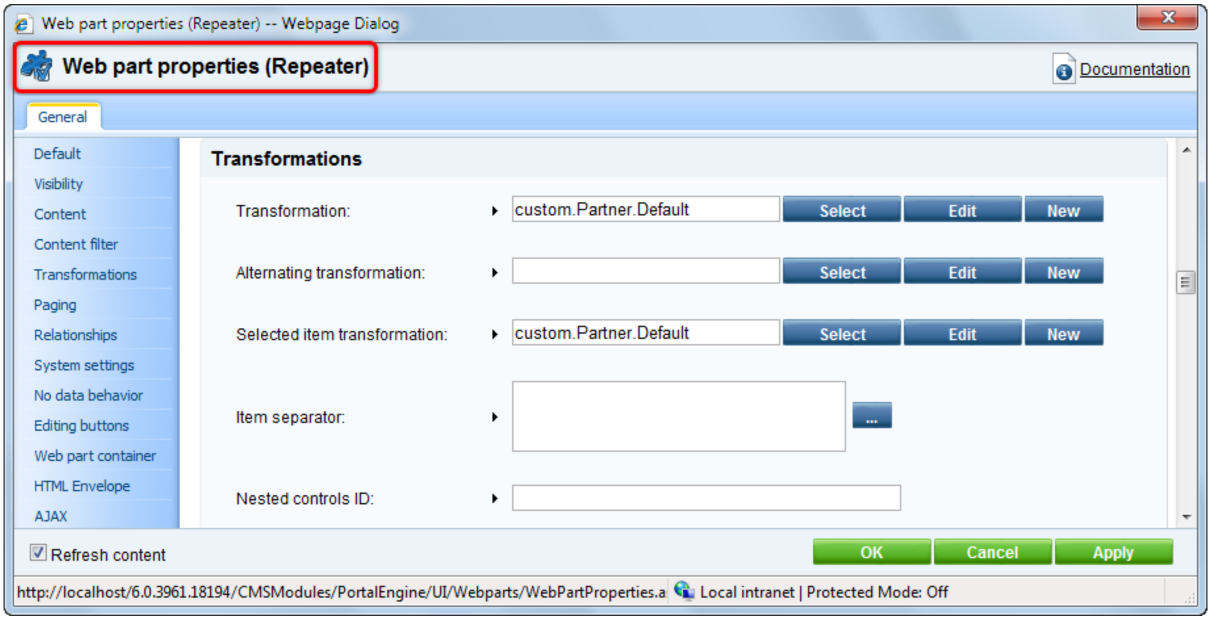
Creating the partner list

1. Go to **CMS Desk -> Content**, choose **Examples** from the content tree, click **New** and choose to create a new **Page (menu item)** document. Call the page **Partner directory** and choose the **Create a blank page** option. Select the **Simple** layout and click **Save**.



2. Switch to the **Design tab** of the newly created page and add the **Listings and viewers/Repeater** web part. Set the following properties:

- **ID:** repeaterPartners
- **Document types:** custom.partner
- **Transformation:** custom.Partner.Default
- **Selected item transformation:** custom.Partner.Default

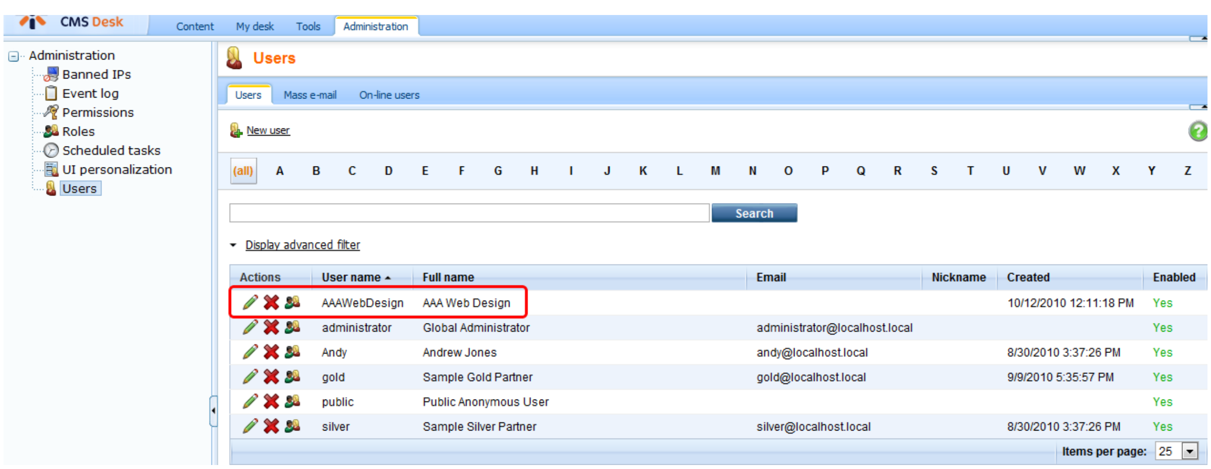


Please note that the default transformation generated by the system will be used. You can later modify the transformation so that it meets your design and layout requirements.

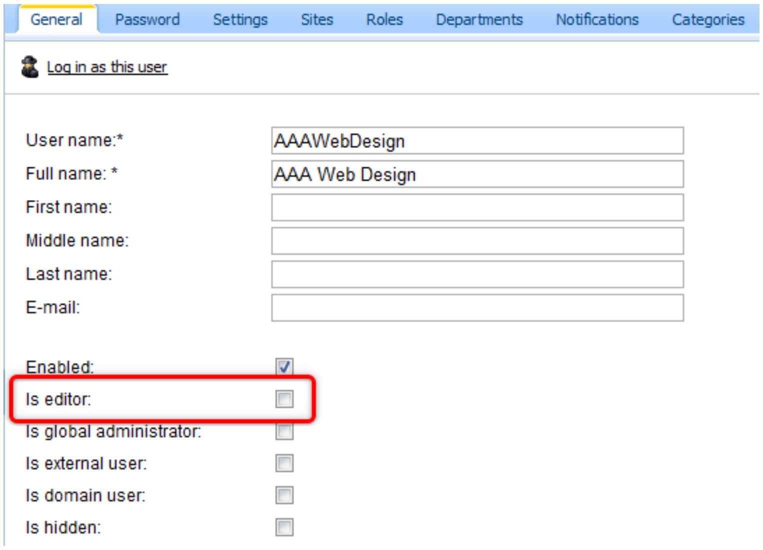
3. Click **OK**. Switch to the **Page** view and you will see an empty page now.

Creating the partner user account

1. Go to **CMS Desk -> Administration -> Users** and by clicking the **New user** link create a new user with user name **AAAWebDesign**. Set the **Is editor** property to no (unchecked) and assign the user to the **CMS Basic users** role.



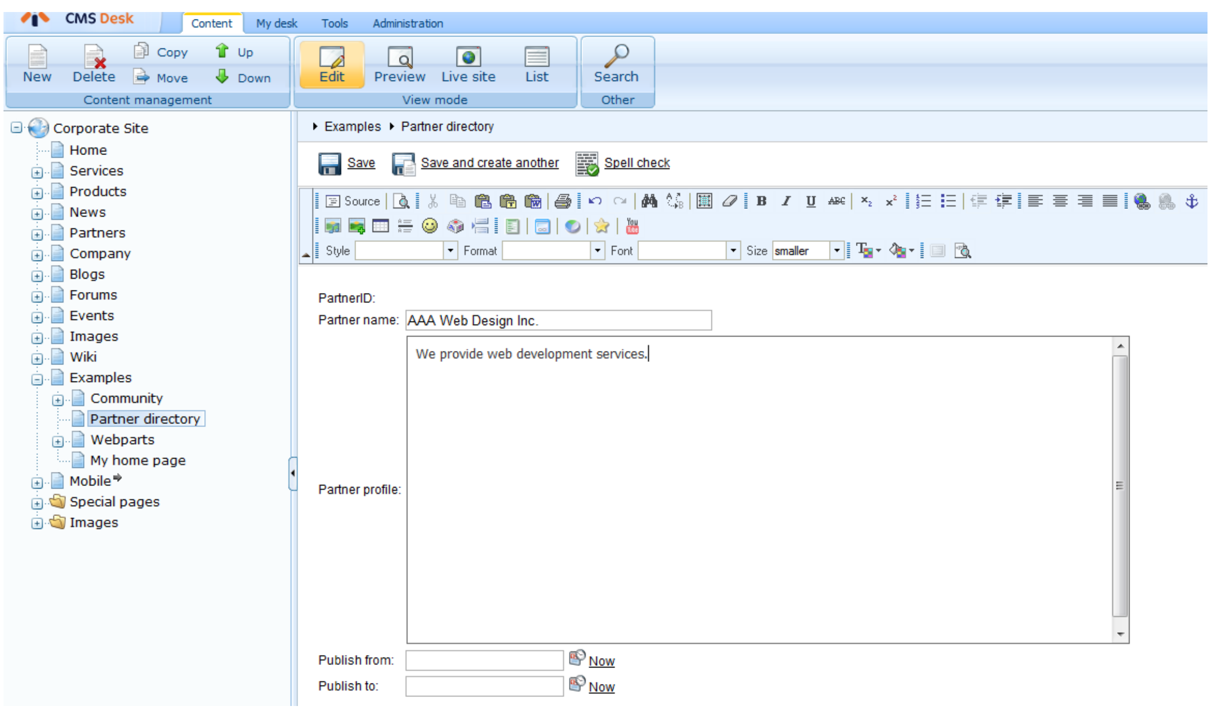
2. For the purpose of this example, choose no (uncheck the box) in the **Is editor** field - **AAA Web Design** is a common site member without access to Kentico CMS Desk. Click **OK** to save the changes.



Creating the partner profile document

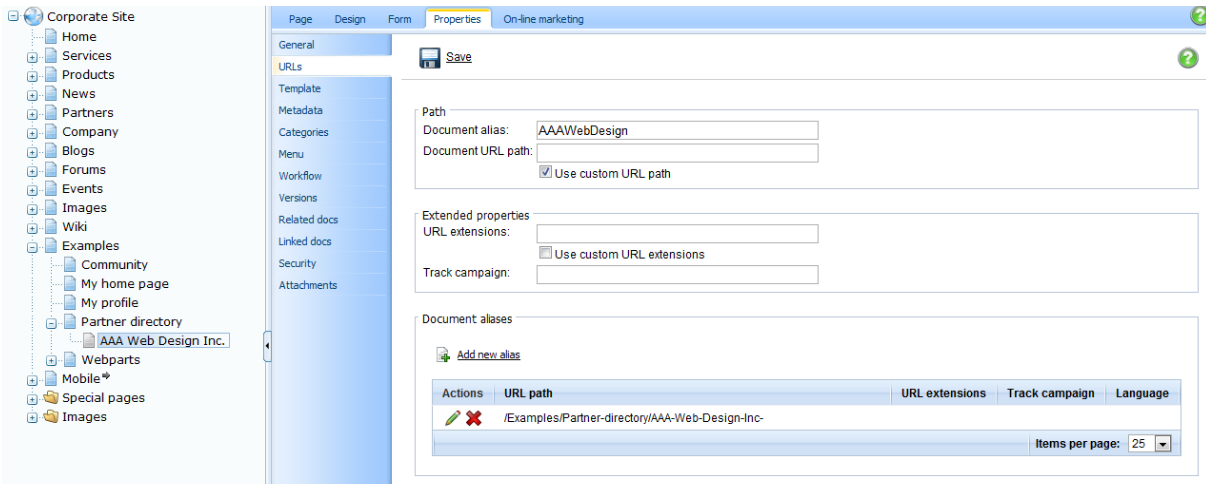
1. Go to **CMS Desk -> Content**, from the content tree, choose **/Examples/Partner directory** and click **New**. Choose to create a new **Partner** and enter the following values:

- **Partner name:** AAA Web Design Inc.
- **Partner profile:** We provide web development services.



2. Click **Save**. Go to the **Properties -> URLs** tab of the newly created document, set the **Document**

alias to **AAAWebDesign** and click **Save**. You need to use the same alias as the user name since you will be using them to match the users to their user profiles. The alias path of the document will always be **`/Examples/Partner-directory/<user name>`**.



3. When you display the page in the **Live site** mode now, you will see a page like this:

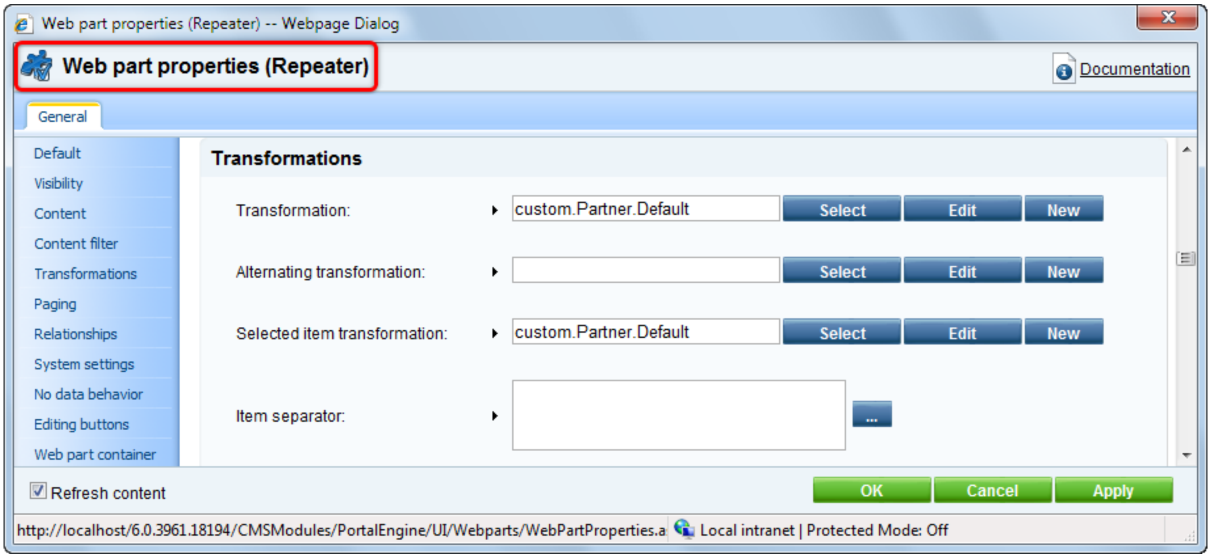
Creating the partner profile editing page

Now you will create a page that will be available only to partners and that will allow them to edit their partner profile.

1. Go to **CMS Desk -> Content**, from the content tree, choose **/Examples** and click **New**. Choose to create a new **Page (menu item)**. Call the page **My profile** and choose the **Create a blank page** option. Choose the **Simple layout** and click **Save**.

2. Switch to the **Design tab** of the newly created page and add the **Listings and viewers/Repeater** web part. It will display the current user's partner profile. Enter the following values and click **OK**.

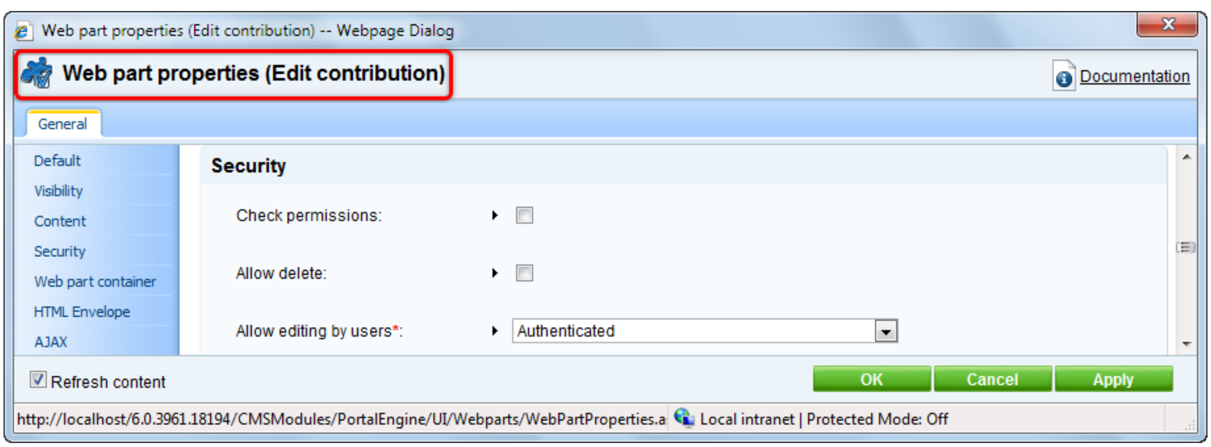
- **ID:** repeaterPartners
- **Path:** /Examples/Partner-directory/{%UserName%}
  - this ensures that the page displays the profile that matches the current user. The {%UserName%} macro is resolved as the user name of the current site visitor.
- **Document types:** custom.partner
- **Transformation:** custom.Partner.Default
- **Selected item transformation:** custom.Partner.Default
- **Hide if no record found:** no (unchecked)
- **No record found text:** No partner profile was found for your user account.
- **Content before:** <h1>Your Partner Profile</h1>



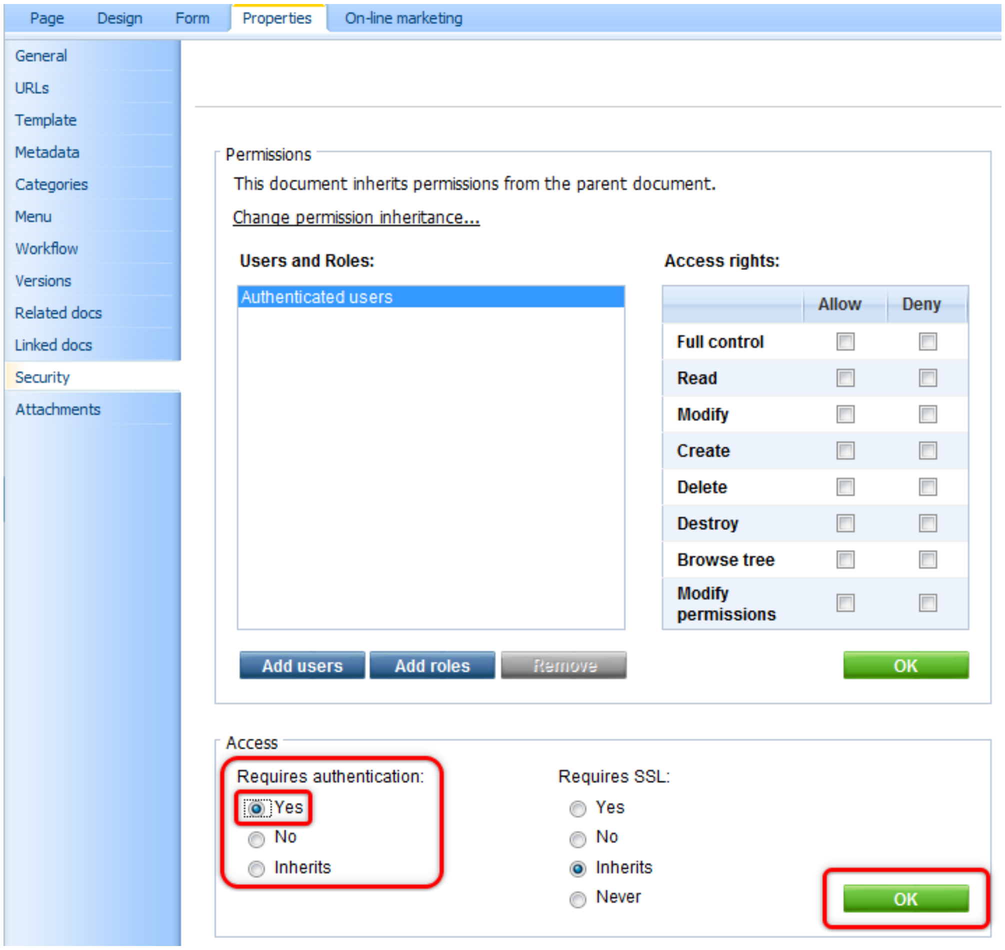
3. Add the **User contributions/Edit contribution** web part, enter the following values and click **OK**.

- **Path:** /Examples/Partner-directory/{%UserName%}
  - this ensures that the form edits the profile that matches the current user. The {%UserName%} macro is resolved as the user name of the current site visitor.
- **Allow delete:** no (unchecked)
  - this option hides the Delete button since you do not want the partners to delete their profile accidentally.
- **Allow editing by users:** Authenticated
  - you want to allow editing only to authenticated users; at the same time, you do not require the user to be a document owner.

Please note that the **Edit contribution** web part is missing the **Allow insert/edit** properties.



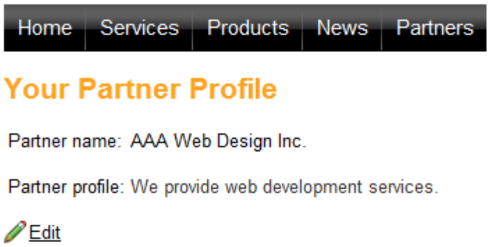
4. As the last step, go to the **Properties -> Security** dialog, set the **Requires authentication** value to **Yes** and click **OK** - the reason is that you want to allow access to this page only to authenticated partners so that you know their user name and can display their profile.



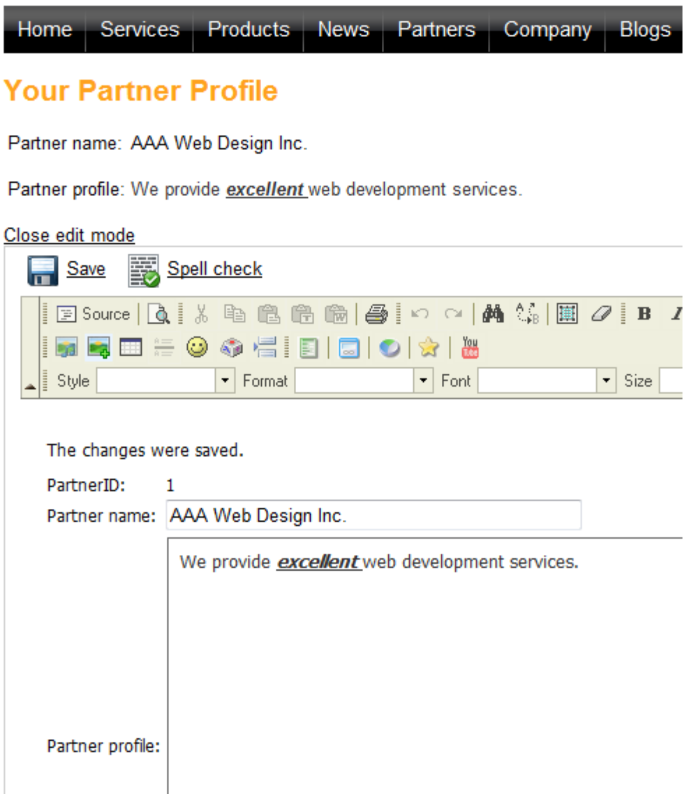
Testing the profile editing page

1. Sign out and go to **Examples -> My profile**. You will be displayed with logon form.

2. Sign in as user **AAAWebDesign** and you will see your partner profile with the **Edit** link.



3. Click **Edit** to open the edit dialog window, modify some values and click **Save**.



4. Go to **Examples -> Partner directory** and you will see the updated profile.

In this topic, you have learnt how to allow users (partners) to edit a single document that matches their user name. It was a little different in comparison to the previous example (Example: Publishing community news), since in this case, you created the document first and only then you mapped it to the user by matching the document alias path and user name. In this example, the partner was not allowed to create new documents and you did not use the document owner security option.

Hint 1: Simplifying the process of creating a new partner

In such cases, it is useful to create a custom module (see Custom modules) that will contain a custom form for creating new partners. Your code will ensure creating the user, the default partner profile and setting the document alias path and user name to the same value. You will need to use Kentico CMS to create the user account (see Managing users) and the partner profile document (see Create document).

Hint 2: Assigning multiple users to the same partner profile

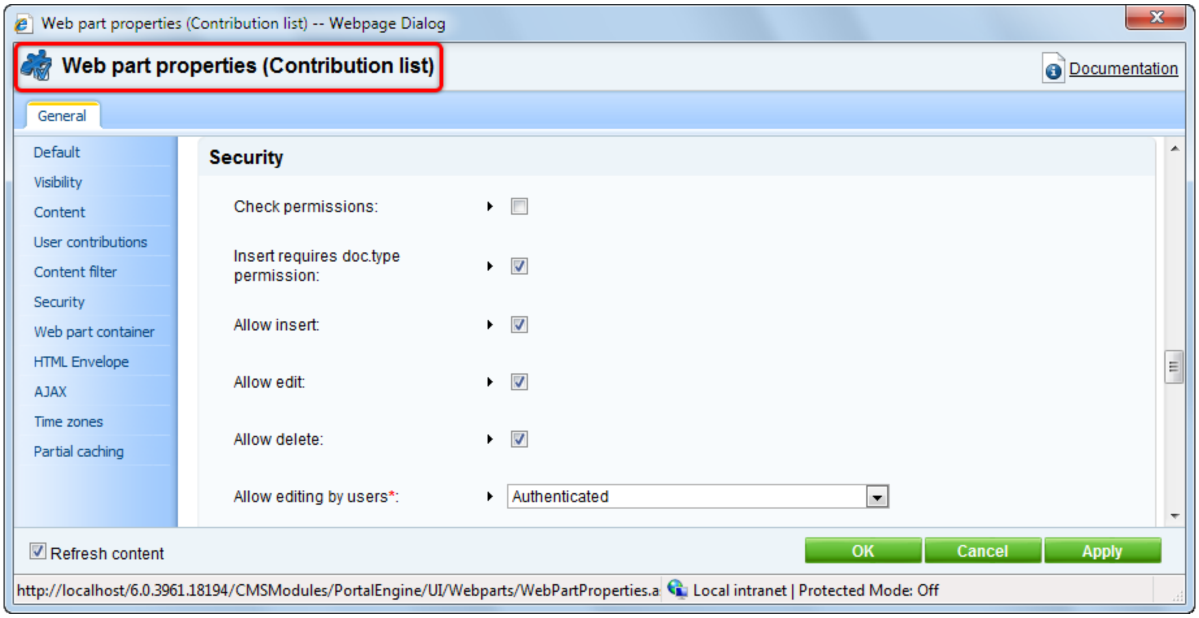
If the user name/document alias path mapping does not fit your needs or if you need to assign multiple users to the same partner profile, you can use another option: Go to **Site Manager -> Development -> System tables -> User** and create a custom text field **PartnerProfileAliasPath**. You will use this custom field in the user profile to specify the partner profile that can be edited by the given user. Then, the **Path** property in the **Edit contribution** web part will be set to value ***{%PartnerProfileAliasPath%}***. Please note that you will find your new custom field on the **Custom Fields** tab, not on the **General** tab as the default attributes.

8.39.4 Security

The User contributions (Wiki) web parts use the following properties to configure their security options:

- **Check permissions** - if you choose this option, appropriate permissions to read/modify/create/delete documents using the User contributions (Wiki) web parts need to be granted to the users. See the Permissions chapter for more details on document permissions.
- **Insert requires doc.type permission** - indicates if document type permissions are required to create a new document.
- **Allow insert/edit/delete** - indicates if the respective buttons should be displayed.
- **Allow editing by users** - you can choose between:

  - **All** - any user who comes to the page with the web part can use it to edit documents
  - **Authenticated** - any authenticated user (site member) can edit the documents; you can use this value in combination with the *NodeOwner = {%CMSContext.CurrentUser.UserID%}* value in the *WHERE condition* property for the web part to display only documents created by the current user (and therefore allow editing of these documents only to them)
  - **Document owners** - only owner of the parent document under which the user contribution documents are stored can edit them



8.39.5 User contributions and API

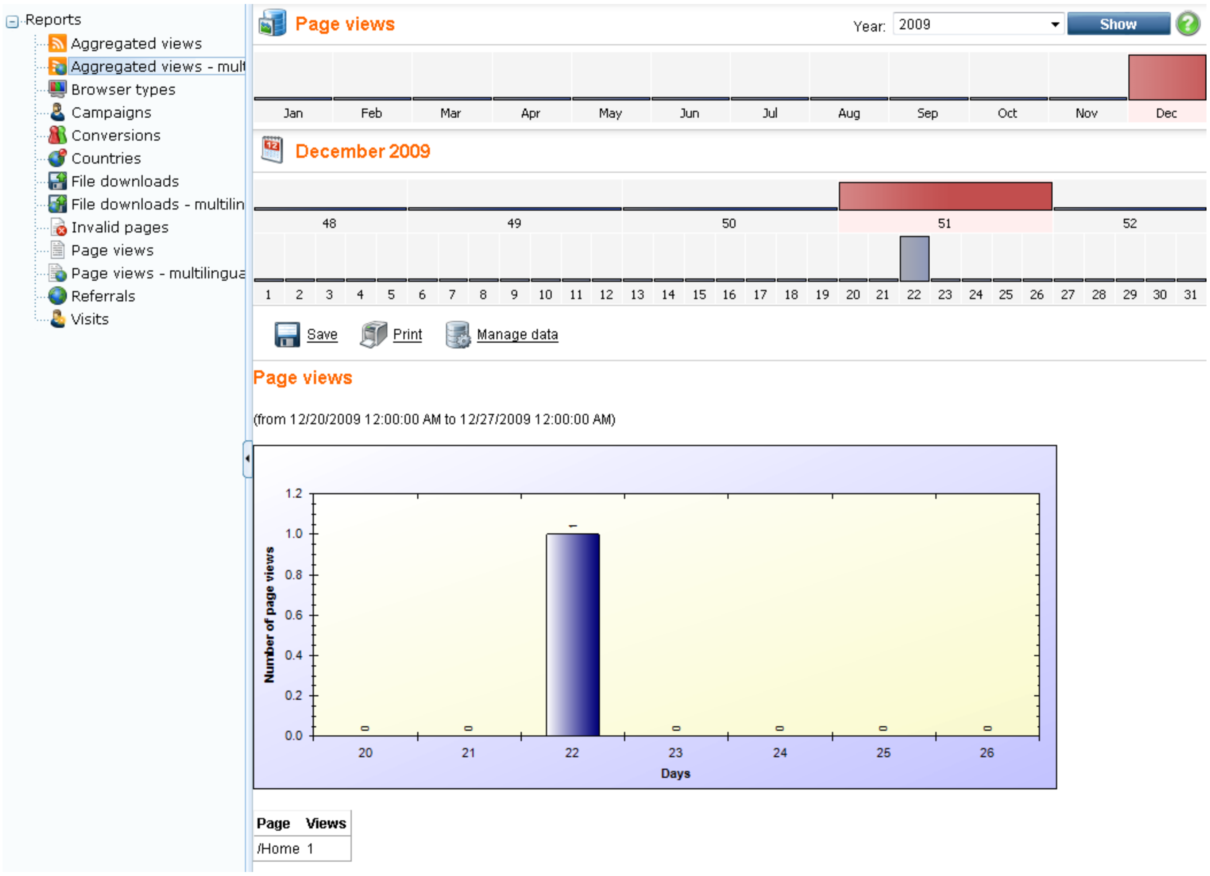
User contributions (Wiki) rely on standard Kentico CMS content management features. As such, they do not have a special programming interface and you need to use the document management API described in the Content management internals chapter.

8.40 Web analytics

8.40.1 Overview

The Web Analytics module allows you to track and analyze website visits, page views, file downloads and other metrics of the website. The module runs in the background of the website collecting all statistics information.

The Web analytics module can be accessed in **CMS Desk -> Tools -> Web Analytics**. As you can see in the screenshot below, when you select a report on the left, the corresponding statistics are displayed on the right. You can choose the year, month, week and day. The screenshot shows a detailed report with graphs and a table.



How it works

When an event (page view, file download, etc.) occurs, it is saved in a file in the *~/App_Data/CMSModules/WebAnalytics/* folder. The format of the file name is:

`<event type code name>_<date>_<time>.log`

Every minute, a new file is created. A scheduled task reads the closed files every minute (or other scheduled interval) and imports the data into the database. This data is then displayed by means of the reports defined in the Reporting module.

The name of the scheduled task is ***Process analytics log*** and you can control its status and settings in **Site Manager -> Administration -> Scheduled tasks**.

Enabling the Web analytics module

The Web Analytics module is disabled by default. You can enable it in **Site Manager -> Settings -> Web Analytics** by checking the **Enable Web Analytics** check-box.

Disk Permissions

Please note that the Web Analytics module requires that the Modify permission for the

~/app_data folder on your disk is granted to the ASP.NET account (see the Disk permissions problems chapter for names of the account under various operating systems).

Limitations

The Web Analytics module tracks only content and events related to pages managed by Kentico CMS. It doesn't track other content, such as html files or media files (jpg, gif, etc.) that are not served by the CMS.

Delay in displaying results

All the events are stored in temporary files and need to be processed on regular basis, by a scheduled task. Please note that there can be a delay between the event (such as page view) and its displaying in the reports.

- To learn with which pre-defined reports the Web analytics module comes, please refer to the Available reports topic.
- To learn about the campaign tracking support in the Web analytics module, please refer to the Tracking marketing campaigns topic.
- To learn about the conversion tracking support in this module, please refer to the Tracking conversions topic.
- To learn how to configure the module, please refer to the Configuration options topic.
- If you would like to learn about the security possibilities of the module, please refer to the Security topic.
- To learn how to create custom reports, please refer to the Creating custom reports topic.

8.40.2 Available reports

The Web Analytics module comes with the following pre-defined reports:

- **Aggregated views** - this report tracks access to pages via links in RSS or Atom feeds created using the Syndication module. All cultural versions of a document are tracked together as one document.

- **Aggregated views - multilingual** - this report tracks access to pages via links in RSS or Atom feeds created using the Syndication module. Each cultural version of a document is tracked separately.

- **Browser types** - this report shows what types of browsers your website visitors use. It recognizes the major and minor version - e.g. FireFox 2.1.

- **Campaigns** - this report helps you track your marketing campaigns - when a visitor comes with a particular URL parameter to your website, you can track the visit from the particular campaign (typically from e-mail or banner campaigns). See Tracking marketing campaigns for more details.

- **Conversions** - this report helps you track conversions on your website. See Tracking conversions for more details.

- **Countries** - this report shows the countries the visitors come from. The countries are recognized by IP address, which may not be a 100% reliable method in all cases, but the overall statistics should provide correct results for a high number of visits.

- **File downloads** - this report summarizes the number of files downloaded by the website visitors and the most downloaded files. Please note that it tracks only the files managed by the CMS and served using the Kentico CMS scripts (GetFile.aspx). All cultural versions of a file are tracked together as one file.
- **File downloads - multilingual** - this report summarizes the number of files downloaded by the website visitors and the most downloaded files. Please note that it tracks only the files managed by the CMS and served using the Kentico CMS scripts (GetFile.aspx). Each cultural version of a file is tracked separately.
- **Invalid pages** - this report summarizes the number of ASPX pages that were not found on the website and the most often requested invalid URLs.
- **Page views** - this report summarizes the number of displayed pages (served by Kentico CMS) and the most often requested pages. All cultural versions of the page are tracked together as one page.
- **Page views - multilingual** - this report summarizes the number of displayed pages (served by Kentico CMS) and the most often requested pages. Each cultural version of a page is tracked separately.
- **Referrals** - this report summarizes the number of visits coming from external websites and the websites from which the visitors come to your website most often.
- **Visits** - this report summarizes the number of visits of your website and the number of new versus returning visitors. The tracking is ensured by a cookie.

8.40.3 Tracking marketing campaigns

In many cases, tracking referring website URLs may not be possible or a reliable way of tracking your on-line marketing campaigns. Then, you may want to use the campaign tracking support in the Kentico CMS Web Analytics module.

How it works

You need to include a special parameter to the link in your campaign that will allow you to identify the campaign name. If your current link that you use for banner advertisement looks like this:

`http://www.mywebsite.com`

your new URL with campaign tracking will look like this:

`http://www.mywebsite.com/`**`?campaign=banner1`**

As you can see, you have added a parameter ***campaign*** with the name of the campaign (banner1).

How to configure campaign tracking

All you need to do is to specify the campaign tracking URL parameter name in **Site Manager -> Settings -> Web analytics**. Choose your website and set the value of the **Campaign tracking URL parameter** key to: ***campaign***

Then, modify the links to your website so that they contain the parameter *campaign* like these links do:

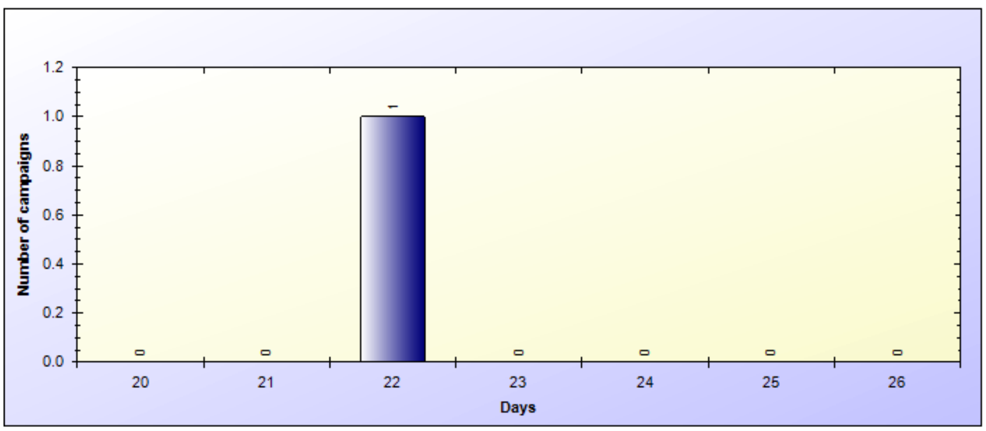
```
http://www.mywebsite.com/?campaign=banner1
http://www.mywebsite.com/?campaign=newsletterJune
http://www.mywebsite.com/?campaign=pressrelease
```

How to evaluate campaigns

Go to the **CMS Desk -> Tools -> Web Analytics** module and choose the **Campaigns** report. You will see how many hits your received from your campaigns in the chosen time period:

Campaigns

(from 12/20/2009 12:00:00 AM to 12/27/2009 12:00:00 AM)



Campaign	Views
banner1	1

8.40.4 Tracking conversions

The conversion tracking allows you to track the behavior of visitors on your website and adjust the website for better results.

Tracking conversions in registration form, newsletter subscription and shopping cart

Kentico CMS comes with built-in support for tracking conversions in the following situations:

User registration

When the user registers, you can track it as a conversion by setting the value **Track conversion name** of the **Registration form** web part to some custom name - e.g. "registration".

Newsletter subscription

When the user registers, you can track it as a conversion by setting the value **Track conversion name** of the **Newsletter subscription** web part to some custom name - e.g. "subscription".

Purchase in the shopping cart

When the user registers or makes a purchase, you can track it as a conversion by setting the value **Registration conversion name** or **Order conversion name** of the **Shopping cart** web part to some custom name - e.g. *cart_registration* or *cart_order*.

Tracking conversions and their campaign or referring websites

If you wish to track the conversion together with campaign name or referring URL, you need to use the conversion name in the following format (for example):

order_{%campaign%}
or
order_{%urlreferrer%}
or
order_{%campaign%}_{%urlreferrer%}

Tracking conversions using your custom code

Using your custom code, you can track a conversion when a visitor makes some action on your website, such as registration, sign up, etc. Then, you can see what changes made to the website helped you increase your conversion rate.

To track a conversion, you need to add a piece of code to your page, web part or to some global event handler:

[C#]

```
string siteName = CMS.CMSHelper.CMSContext.CurrentSiteName;
if (CMS.WebAnalytics.AnalyticsHelper.AnalyticsEnabled(siteName)
  && !CMS.WebAnalytics.AnalyticsHelper.IsIPExcluded(siteName, HttpContext.Current.Re
{
   // log conversion, the conversion name is "order"
   CMS.WebAnalytics.HitLogProvider.LogHit("conversion", siteName, null, "order", 0);
}
```

Then, the conversion is tracked and you can see the statistics in the **Conversions** report.

You may want to use the following properties of the current user that provide information on the website and/or campaign from which the user came:

[C#]

```
using CMS.CMSHelper;

// the following values provide the information from user's cookie
CMSContext.CurrentUser.URLReferrer //website URL from which the visitor came
CMSContext.CurrentUser.Campaign //campaign from which the visitor came

// after the user registers, the information is available in the following database-
```

```
CMSContext.CurrentUser.UserURLReferrer //website URL from which the visitor first ca
CMSContext.CurrentUser.UserCampaign //campaign from which the visitor first came
```

8.40.5 Configuration options

You can configure several options of the Web Analytics module in **Site Manager -> Settings -> Web Analytics** dialog:

Enable web analytics	If true, the web analytics module logs the events.
Exclude search engines	Indicates if search engine robots should be excluded from tracking.
Excluded URLs	URLs of your websites that will not be tracked. All underlying pages will not be tracked either. You can specify multiple URLs (or site sections) separated by a semicolon (;).
Excluded file extensions	Types of files managed by Kentico CMS that will not be tracked and shown in the File downloads report. You can use several extensions separated with a semicolon (;) (e.g. .jpg;.gif). Please note: it's necessary to include the dot in the extension name.
Excluded IP addresses	Client IP addresses the access of which will not be tracked by the analytics module separated by a semicolon (;). You can use it to exclude your own IP address so that your visits do not influence the results. Please note that you can use * in IP addresses as a wildcard character that stands for any digit.
Remember visitors by IP (minutes)	If the entered value is higher than 0, the visitors' IP addresses are used for smarter visits logging. IP addresses are stored in memory for the specified amount of time.
Track browser types	Enables browser types tracking.
Track campaigns	Enables campaigns tracking.
Campaign tracking URL parameter	URL parameter name to identify the campaign.
Track conversions	Enables conversions tracking.
Track countries	Enables countries tracking.
Track file downloads	Enables file downloads tracking.
Track invalid pages	Enables invalid pages tracking.
Track page views	Enables page views tracking.
Track referrals	Enables referrals tracking.
Track visits	Enables visits tracking.
Track registered users	Enables registered users tracking.

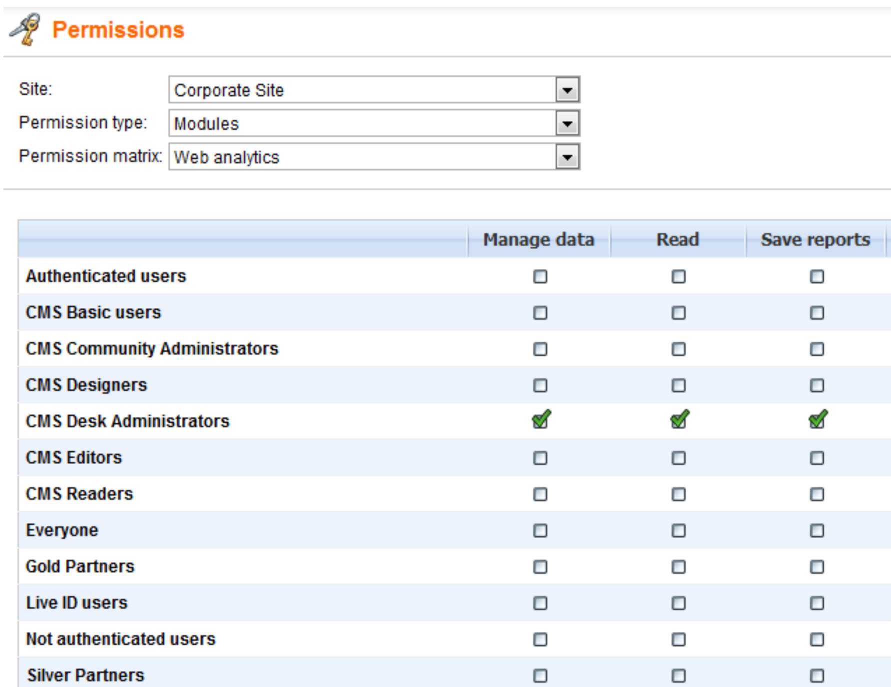
Please note: These options are applied at the moment when the event occurs. A later change will modify only future events.

8.40.6 Security

You can configure access to the Reporting module in **CMS Desk -> Administration -> Permissions**. Choose the permission matrix for **Modules -> Web Analytics**.

The following permissions can be given to the listed roles:

- **Manage data** - allows members of the role to manage Web Analytics data (use the Manage data button)
- **Read** - allows members of the role to read Web Analytics reports
- **Save reports** - allows members of the role to save Web Analytics reports; the saved reports can be viewed in the **Reporting** module



	Manage data	Read	Save reports
Authenticated users	☐	☐	☐
CMS Basic users	☐	☐	☐
CMS Community Administrators	☐	☐	☐
CMS Designers	☐	☐	☐
CMS Desk Administrators	☑	☑	☑
CMS Editors	☐	☐	☐
CMS Readers	☐	☐	☐
Everyone	☐	☐	☐
Gold Partners	☐	☐	☐
Live ID users	☐	☐	☐
Not authenticated users	☐	☐	☐
Silver Partners	☐	☐	☐

8.40.7 Creating custom reports

This example describes in detail how to log custom events and how to display result statistics. A button click event will be logged for this purpose.

Logging the event

To log custom events for web analytics, use the following code from *CMS.Analytics* namespace:
HitLogProvider.LogHit(string codeName, string siteName, string culture, string objectName, int objectId [, int count]);

- **codeName** - statistics code name, it is also used in report's code names
- **siteName** - site name
- **culture** - culture code

- **objectName / objectId** - it is possible to specify the log either by name or by ID
- **count** - it is weight of the log, default value is 1

An example of logging a button click event follows:

[C#]

```
using CMS.WebAnalytics;
using CMS.CMSHelper;

protected void Button1_Click(object sender, EventArgs e)
{
  if (AnalyticsHelper.AnalyticsEnabled(CMSContext.CurrentSiteName))
  {
    HitLogProvider.LogHit("buttonclicked", CMSContext.CurrentSiteName, null, CMSCont
  }
}
```

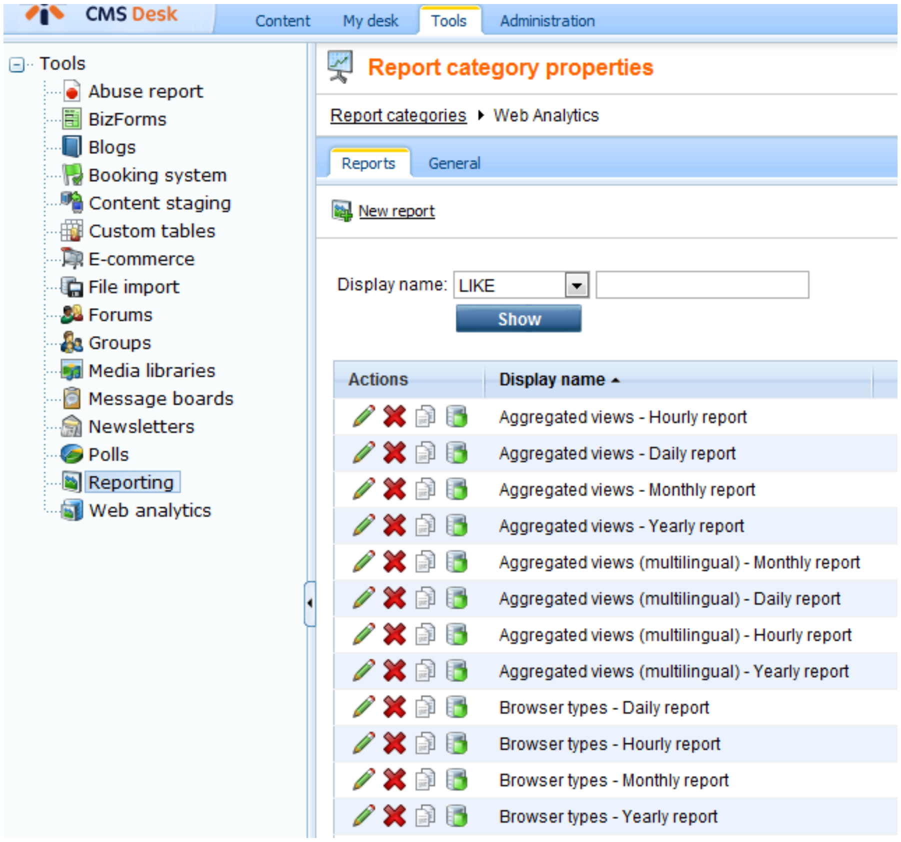
The statistics code name is *buttonclicked* and the logged value is user name of the current user who clicked the button. Now logs will be created and stored in the database after clicking the button. Before you can see your statistics in **CMSDesk -> Tools -> Web Analytics**, the reports have to be done in **CMSDesk -> Tools -> Reporting**. The next step is creation of reports which display statistics in a graph or table form.

Creating reports

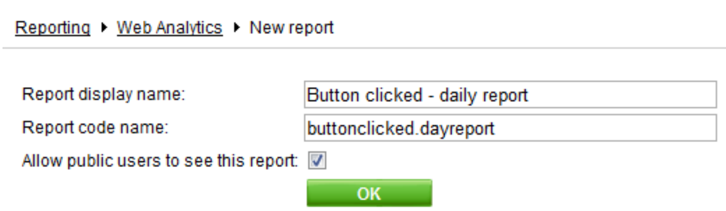
Go to **CMSDesk -> Tools -> Reporting** and choose to **Edit** () the **Web Analytics** category. There are some reports already in the list. These reports belong to default statistics like page views, visits, campaigns, etc. Each statistics has four reports: daily, weekly, monthly and yearly report.
Code names of the reports have to be in a specific form:

- <statisticscodename>.yearreport
- <statisticscodename>.monthreport
- <statisticscodename>.dayreport
- <statisticscodename>.hourreport

In our example <statisticscodename> is *buttonclicked*, as defined above.

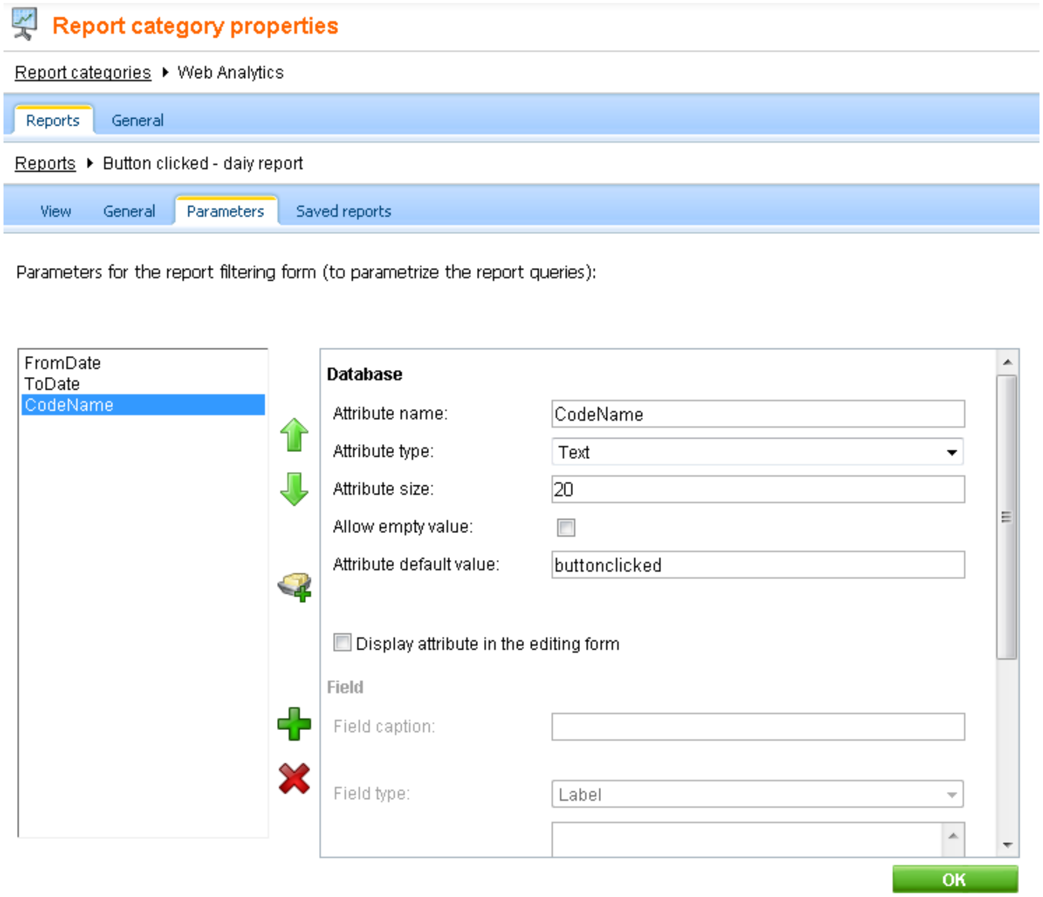


Now you will create a daily report for the new statistics. Click the **New report** link and enter the values as displayed in the screenshot below. Please do not forget to check the **Allow public users to this report** checkbox.



Go to the **Parameters** tab and by clicking the **New attribute** () icon create three essential parameters which will be used in queries:

- **FromDate -** *Date and Time* attribute type
- **ToDate** - *Date and Time* attribute type
- **CodeName** - *Text* attribute type; statistics code name as attribute default value

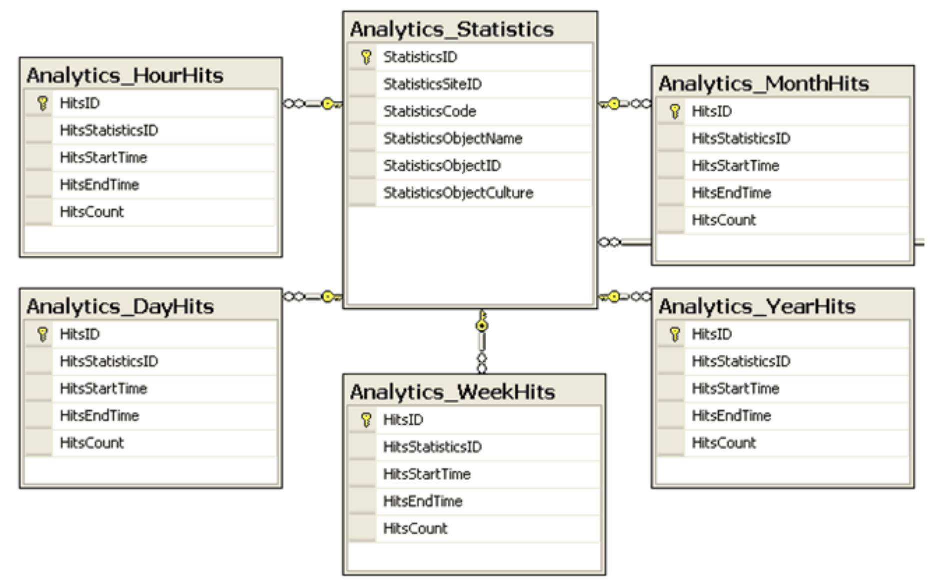


Disable the Display attribute in the editing form for all parameters.

Go to the **General** tab and click the **Add** button in the **Tables** section to create a report table:

Please note that there are six important database tables used in the web analytics module.

The *Analytics_Statistics* table contains general information about statistics, other five tables store statistics hit records for a specific time period – *Analytics_HourHits*, *Analytics_DayHits*, *Analytics_WeekHits*, *Analytics_MonthHits* and *Analytics_YearHits*. As the report is a daily report, the *Analytics_DayHits* table is used together with the *Analytics_Statistics* table. See the diagram below to understand the database structure of web analytics:

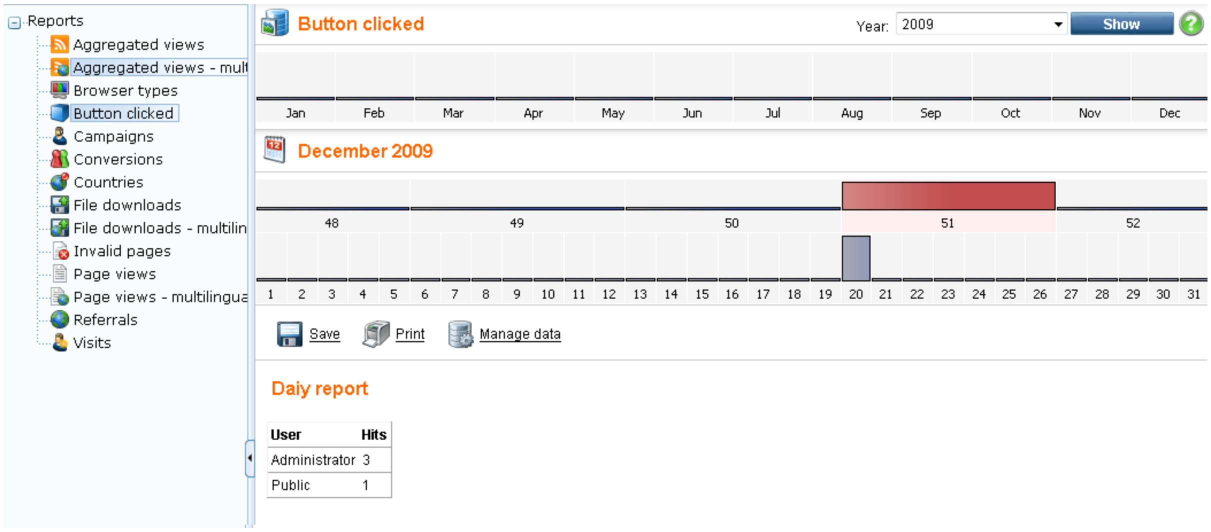


Now click **OK** and use the **Insert** button in the **Tables** section to insert a macro into the layout of the report. Click **OK** again - the daily report is now ready.

1. Other reports are similar, the only difference is in queries where you should use the appropriate tables like *Analytics_YearHits* in the yearly report, *Analytics_MonthHits* in the monthly report, etc.

2. The last thing you should do is insert a string into resources, which will be used in the statistics list in **CMSDesk -> Tools -> Web Analytics**. The form of the resource key has to be as follows – **analytics_codename.<codename>**; it is *analytics_codename.buttonclicked* for the purpose of this example. You may also add a specific icon for new statistics. Icons are stored in **~\App_Themes\<used theme>\Images\CMSModules\WebAnalytics\Icons**. The name should be in form **<codename>.gif**. If there are any '.' in codename, replace them with '_'.

And finally the result:



8.41 WebDAV integration

8.41.1 Overview

WebDAV stands for *Web-based Distributed Authoring and Versioning*. It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers. Detailed information, specifications and documentation related to this technology can be found on its official website: http://www.webdav.org/. Other valuable information is located on its Wikipedia page: http://en.wikipedia.org/wiki/WebDAV.

WebDAV integration in Kentico CMS enables users to open, edit and save document attachments or documents using the File field (e.g. *CMS.File* document type) in a client application installed on their local computer (e.g. *Microsoft Office Word*, *Excel*, *PowerPoint*, *MS Paint*, etc.). The advantage of this approach is that no local copy of the edited document needs to be created and uploaded to the server manually after editing. Saving the document in the client application uploads the new version to the server automatically. This provides a much more straightforward process of editing attached documents, especially for non-experienced or non-technical end users.

The Requirements topic gives an overview of which software and configuration is required for WebDAV integration to be functional both on the server- and the client-side. The Configuration for WebDAV topic explains how a website needs to be installed and configured in order to meet these requirements.The Editing files using WebDAV topic explains the process of WebDAV editing in a simple example.

WebDAV editing has been integrated into the User contributions and Document library modules, which provides WebDAV editing even to users on the live site. This possibility is explained in the Integration with User contributions and Integration with Document library topics. WYSIWYG editor dialogs also provide built-in WebDAV editing support, as described in Integration with WYSIWYG editor dialogs.

WebDAV editing also fully integrates with workflow and versioning. When there is a workflow defined for a document and it or its attachment is edited using WebDAV, new version of the document is created, check-in/check-out is performed, etc., based on settings of the particular workflow.

8.41.2 Requirements

The following requirements must be met on the server- and the client-side in order for WebDAV integration to be functional in Kentico CMS:

Server-side requirements

- **Windows authentication** must be enabled in IIS and the CMS must be configured to use this type of authentication.
- If you are using an older version of IIS than 7.0, the Kentico CMS website must be **installed in the root** of the server. This is necessary because Microsoft Office submits configuration requests to the site root and requires the server to respond properly. If the website is not installed to the root, Microsoft Office will open its associated documents as read-only.
- If you are using IIS 7.0 or higher, the website needn't be installed in the root - it can also be installed in a virtual directory.

Client-side requirements

- WebDAV can be used only in **Internet Explorer 6 or higher**. No other web browsers are currently supported.
- **Client applications with WebDAV support** for a particular file types need to be installed on the client machine. E.g. *Microsoft Office 2003* or higher or *Paint* and *Notepad* in *Windows 7* support editing of associated file types using WebDAV.
- **Matching (x86) or (x64) versions** of both *Internet Explorer* and *Microsoft Office* must be installed. In other words, to be able to open documents using WebDAV in *Internet Explorer (x86)*, *Microsoft Office (x86)* must be installed on the client computer. For *Internet Explorer (x64)*, you need to have *Microsoft Office (x64)* installed.

The Configuration for WebDAV topic explains how the required configuration can be achieved.

8.41.3 Configuration for WebDAV

The following steps need to be taken in order to install a Kentico CMS website with WebDAV support enabled. The beginning of the procedure is different for IIS 7.0 or higher and the previous versions of IIS.

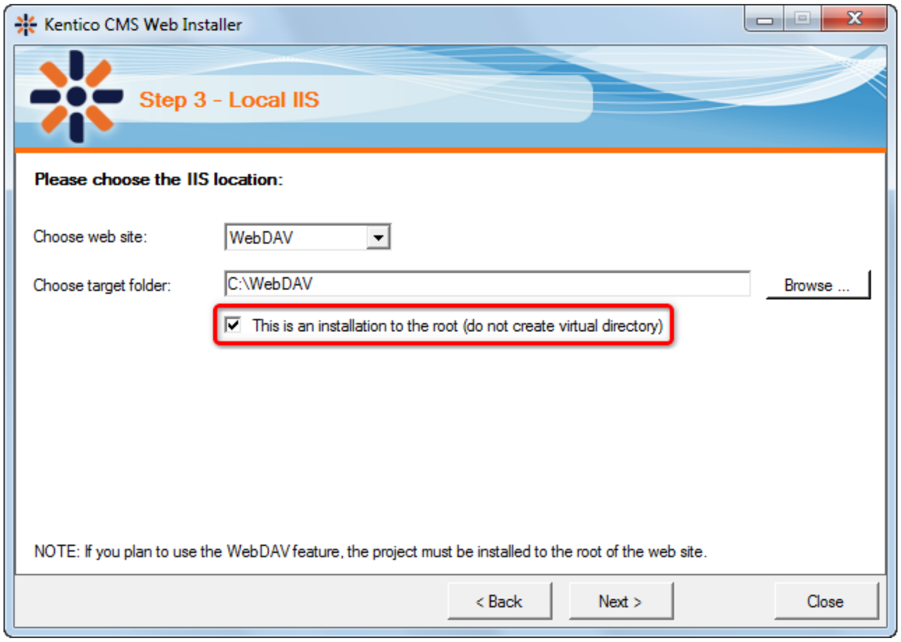
Step 1 on IIS versions prior to 7.0

Before you start, it is necessary to have a website created in the IIS root (i.e. not in a virtual directory). The website should have *Windows authentication* enabled in IIS.

1. Launch Kentico CMS Web installer. In step 2, choose **I want to use local IIS server**. In step 3, use the following configuration:

- **Choose web site**: choose the website that you have prepared in the IIS root
- **Choose target folder**: enter the path to the website's physical folder
- **This is an installation to the root (do not create virtual directory)**: enabled

Click **Next** and proceed through the rest of the wizard.

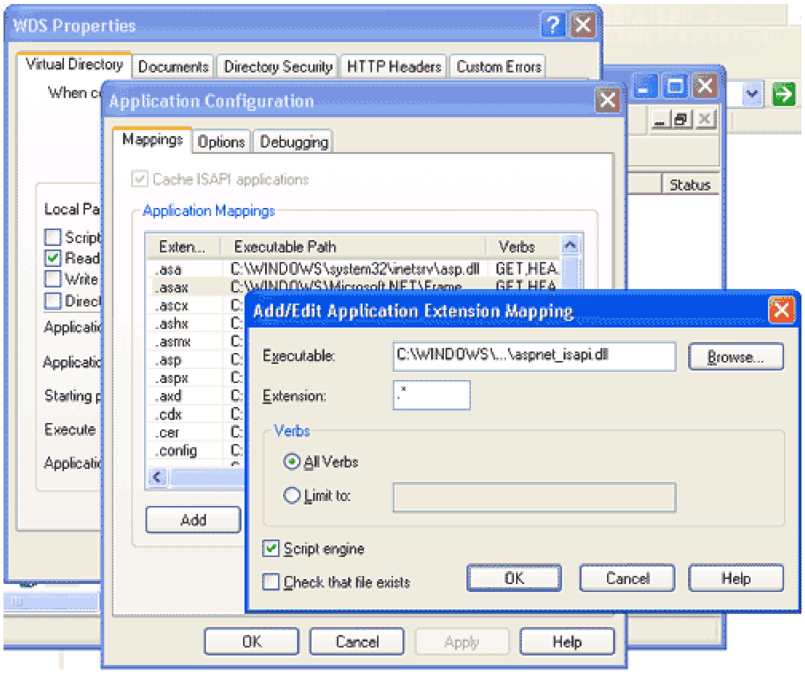


When using IIS versions older than 7.0, it is also necessary to correctly map the **aspnet_isapi.dll** file in IIS. The process of performing this configuration is described below.

Configuration for IIS 5 and earlier versions

Open the IIS Management console, right-click on your Kentico application in the tree and select **Properties**. On the **Virtual Directory** tab, click **Configuration** and then **Add** to create a new application extension mapping. Enter the following details into the dialog:

- **Executable**: <Windows install directory>\Microsoft.NET\Framework\<.Net framework version>\aspnet_isapi.dll
- **Extension**: .*
- **Verbs**: All Verbs

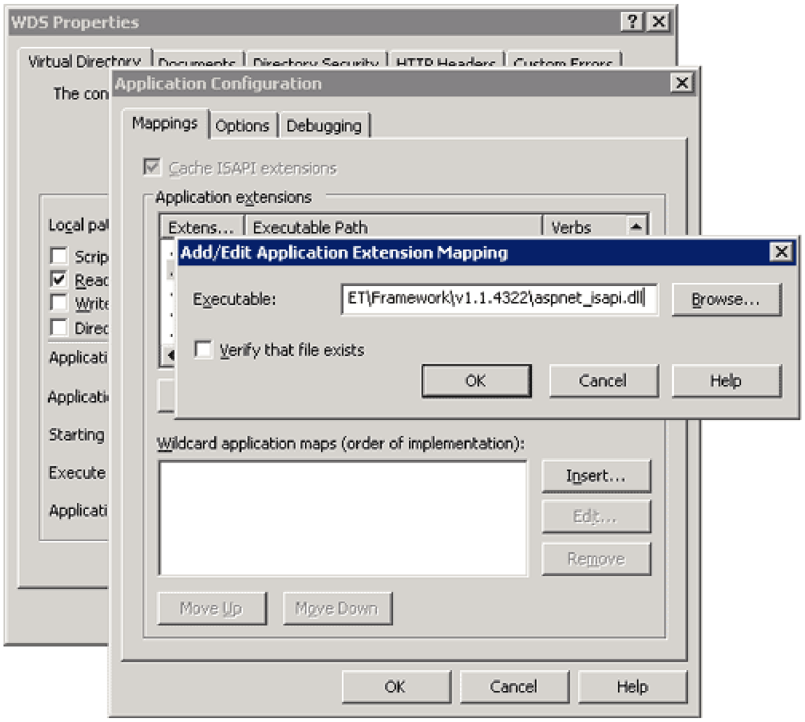


Click **OK** and return to the main management console. Now right-click the website under which your application is stored and select **Properties**. Switch to the **Home directory** tab, ensure that the **Read** box is checked and set the **Execute permissions** field to *Scripts only*.

Configuration for IIS 6

Open the IIS 6 Management console, right-click on your Kentico application in the tree and select **Properties**. On the **Virtual Directory** tab, click **Configuration** and then **Insert** to add a new wildcard application map. Enter the following details into the dialog:

- **Executable**: <Windows install directory>\Microsoft.NET\Framework\<.Net framework version>\aspnet_isapi.dll



Click **OK** and return to the main management console. Now right-click the website under which your application is stored and select **Properties**. Switch to the **Home directory** tab, ensure that the **Read** box is checked and set the **Execute permissions** field to *Scripts only*.

Step 1 on IIS 7.0 or higher

On IIS 7.0 or higher, you do not need to have the website created in the IIS root - it can also be installed in a virtual directory.

1. Launch Kentico CMS Web installer and proceed through the steps with configuration according to your needs.

If you wish to run the website using an application pool set for *Classic* managed pipeline mode, the two Isapi extension handlers highlighted below must be added to your **web.config** file once the **Web installer** finishes:

```
...
<!-- WebDAV location BEGIN -->
<location path="cms/files">
  <system.web>
    <httpHandlers>
      <clear />
      <add verb="*" path="*" type="CMS.WebDAV.WebDAVHandler, CMS.WebDAV" />
    </httpHandlers>
    <httpRuntime executionTimeout="2400" maxRequestLength="2097151" />
  </system.web>
  <system.webServer>
    <handlers>
      <clear />
```

```
      <add name="aspnet_isapi 32-bit" path="*" verb="*" modules="IsapiModule"
      scriptProcessor="%windir%\Microsoft.NET\Framework\v2.0.50727\aspnet_isapi.
      dll" resourceType="Unspecified" requireAccess="Script" preCondition="
      classicMode,runtimeVersionv2.0,bitness32" />
      <add name="aspnet_isapi 64-bit" path="*" verb="*" modules="IsapiModule"
      scriptProcessor="%windir%\Microsoft.NET\Framework64\v2.0.50727
      \aspnet_isapi.dll" resourceType="Unspecified" requireAccess="Script"
      preCondition="classicMode,runtimeVersionv2.0,bitness64" />
      <add name="CMSWebDAVHandler" path="*" verb="*" type="CMS.WebDAV.
WebDAVHandler, CMS.WebDAV" />
    </handlers>
    <security>
      <requestFiltering>
        <requestLimits maxAllowedContentLength="2147483648" />
      </requestFiltering>
    </security>
  </system.webServer>
</location>
<!-- WebDAV location END -->
...
```

If necessary, adjust the path in the **scriptProcessor** attributes of the handlers according to your current . Net framework version.

The rest of the configuration - common for all IIS versions

2. When the web installation is complete and the appropriate configurations have been made according to your IIS version and application pool mode, open the website using the link in the last step of the **Web installer**. When you access the URL, **Kentico CMS Database Setup** will be displayed. Follow the wizard as described in Installation and deployment -> Installation procedure -> Database setup.

3. After finishing the Database Setup, go to **Site Manager -> Settings -> WebDAV** and configure the settings as described in the Settings topic.

4. The last step is to enable **Windows authentication** for your website. Please follow the instructions in Development -> Membership -> Authentication -> Windows authentication -> Configuring Windows authentication.

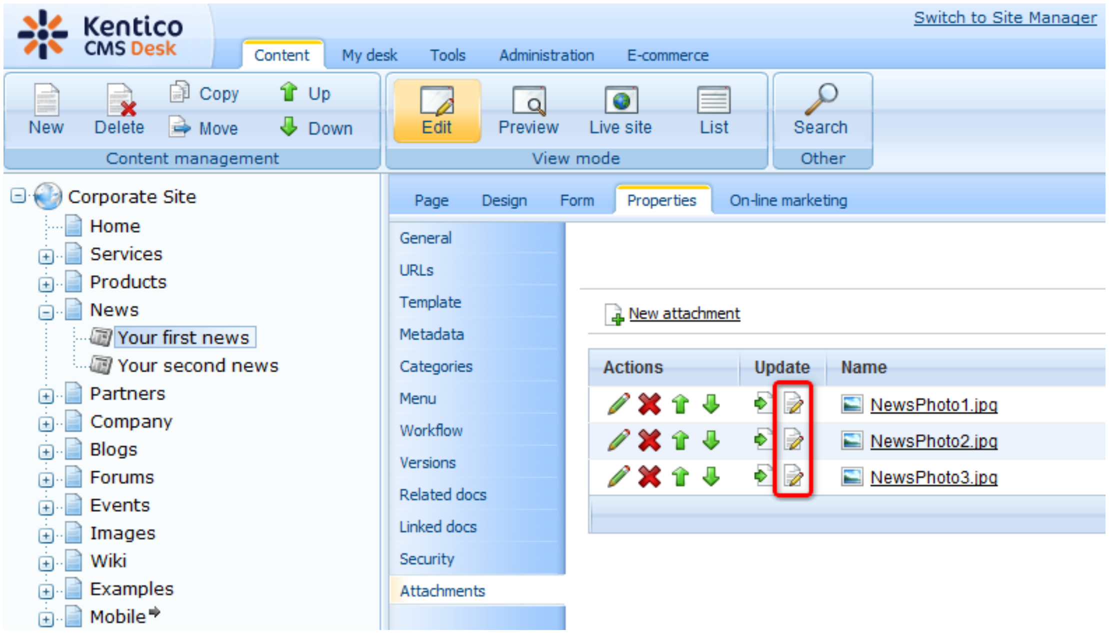
With all these steps performed, WebDAV editing should be possible. Please refer to the Editing files using WebDAV topic to learn more about the editing possibilities.

8.41.4 Editing files using WebDAV

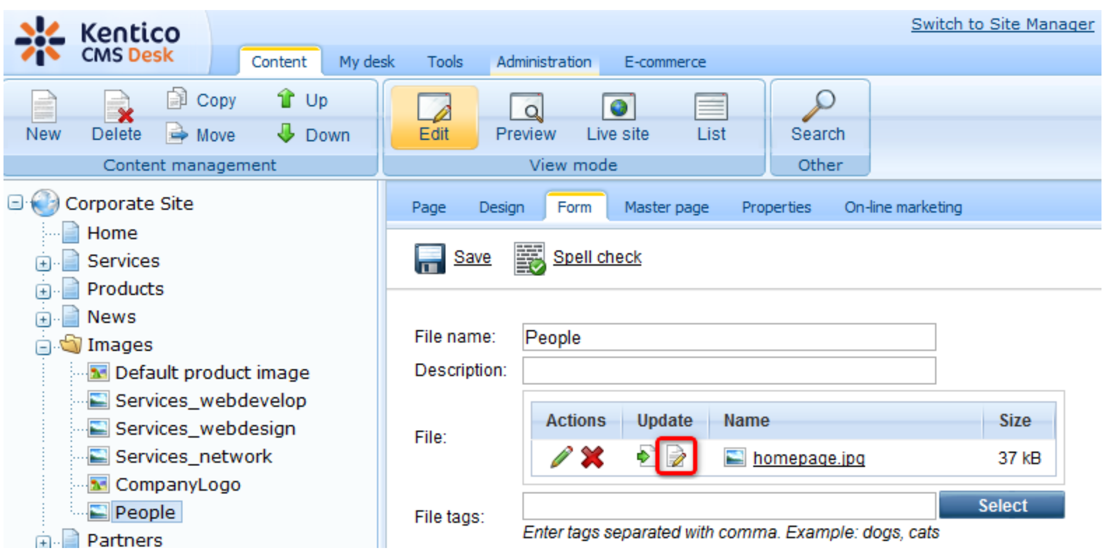
After configuring your server as described in Configuration for WebDAV, users who meet the client-side requirements listed in the Requirements topic may start editing document attachments or documents using the File field (e.g. *CMS.File* document type) using WebDAV.

1. WebDAV editing of document attachments is possible in **CMS Desk -> Content -> Edit**, after selecting a document in the content tree and switching to its **Properties -> Attachments** tab. The **Edit in client application (** **)** should be displayed next to each attachment whose extension is included in **Site Manager -> Settings -> WebDAV -> Supported file extensions**.

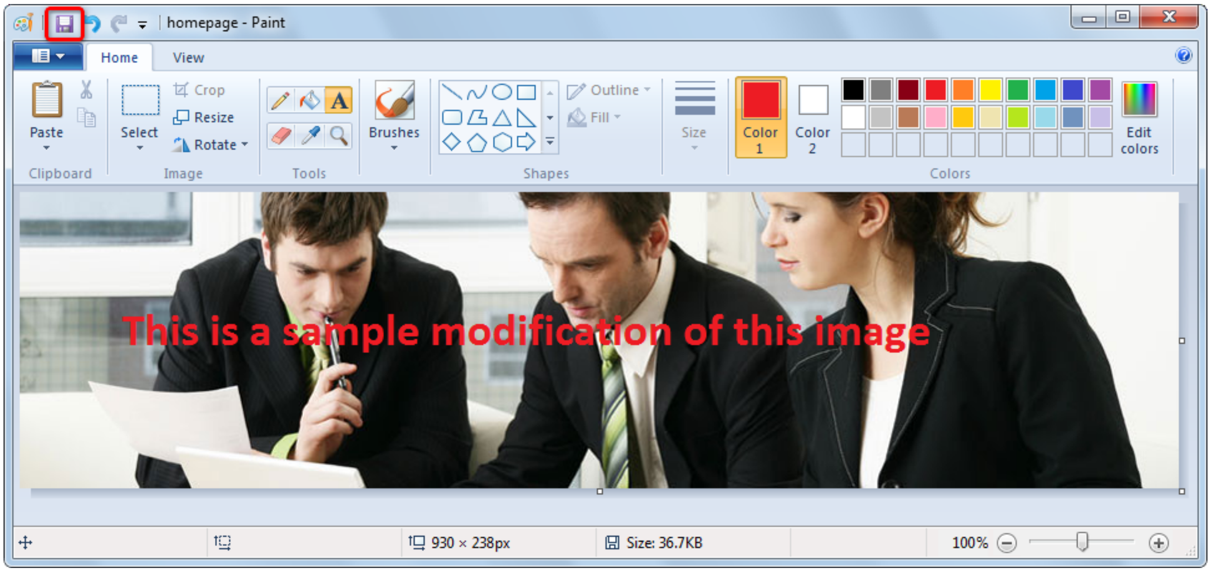
For the icon to be enabled, the **Modify** permission for the current document must be granted to the current user (or one of their roles) on one of the three levels described in Development -> Membership -> Permissions -> Document permissions.



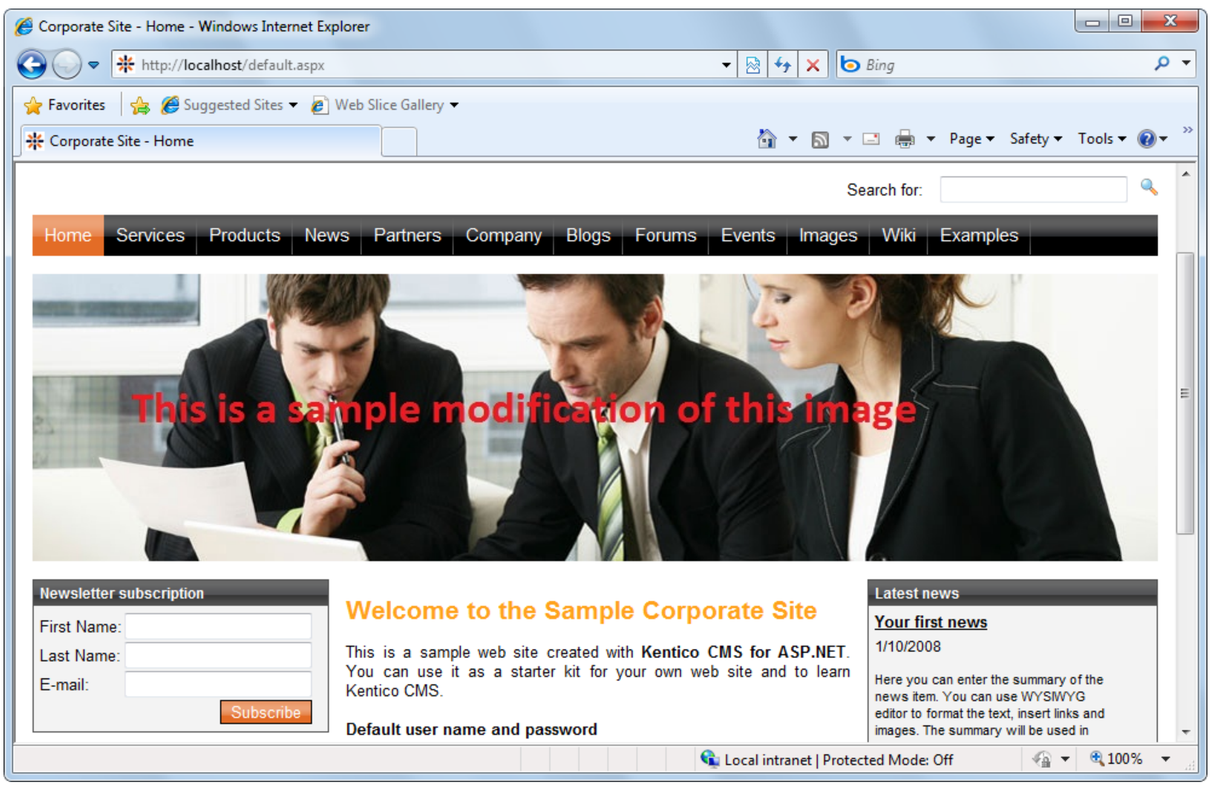
2. The same **Edit in client application (** **)** icon is displayed on the **Form** tab of documents using the File field (e.g. *CMS.File* document type), where a file with a supported extension is uploaded.



3. Clicking the icon opens the document in a client application associated with the particular file extension (e.g. images are opened in MS Paint in Windows 7 by default). Try editing the opened file and save your changes using the **Save** button.



4. Now when you access the live site, you should see that the file that you modified has been saved to the server and that the updated version is used on the live site.

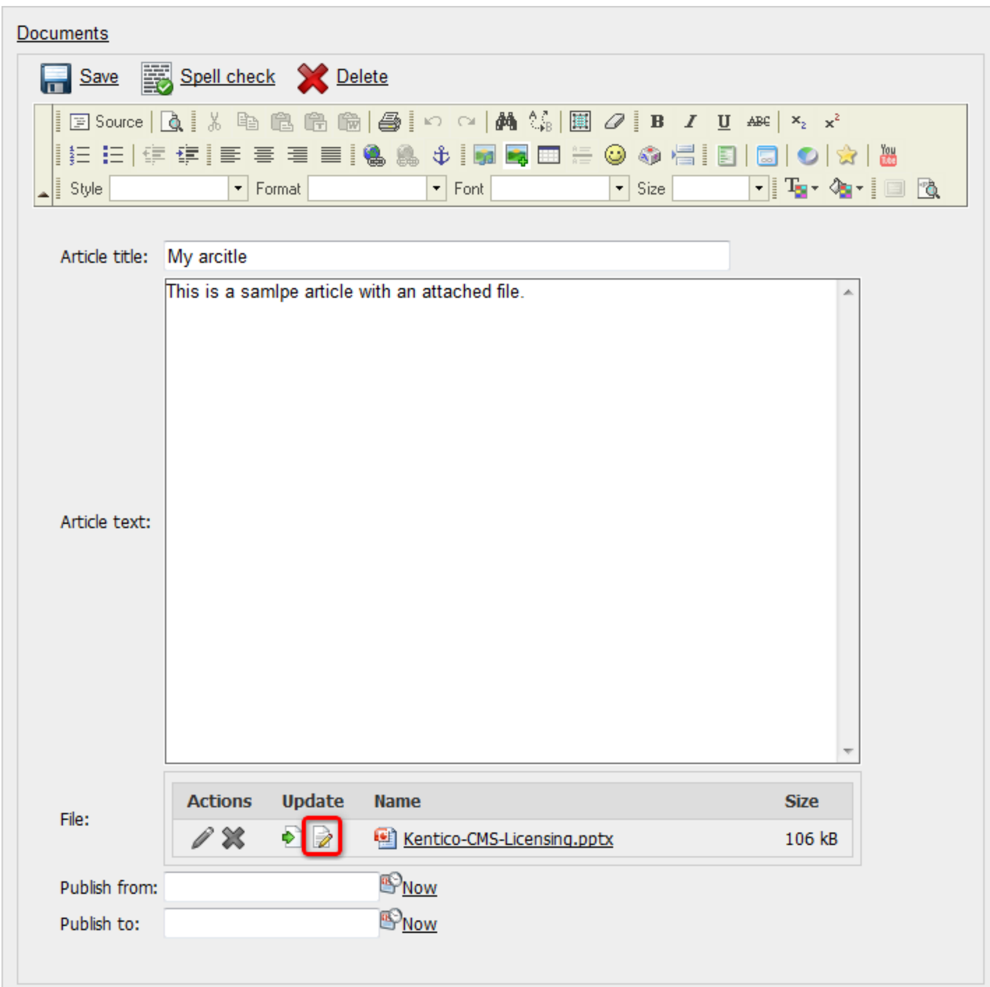


You have learned how document attachments or documents using the File field can be edited using a client application when WebDAV integration is enabled. Editing of any other file types in any other client application (e.g. *Microsoft Office Word*, *Excel*, etc.) can be performed exactly the same way - all you need to do is click the **Edit in client application (** **)** button, edit the file and save it in the client application. Transfer of the updated file to the server is handled by the CMS automatically.

8.41.5 Integration with User contributions

WebDAV editing is integrated into the the User contributions module, which brings WebDAV editing to the live site. If a document type used for user contributions contains a field of the **File** or **Document attachment** type, the **Edit in client application ()** icon is displayed with files uploaded into these fields. Clicking the icon opens the document for editing in a client application (if there is an application for the particular file type installed on the client machine and if the application supports WebDAV).

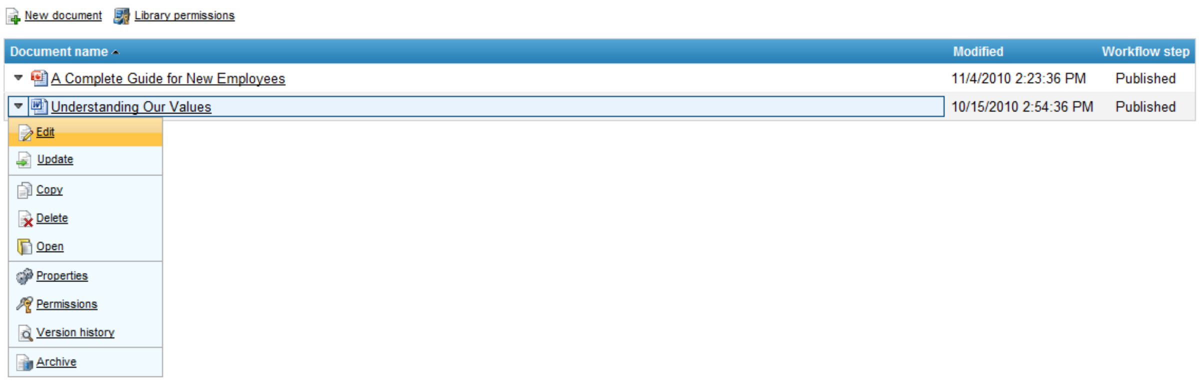
For the icon to be displayed in the **Contribution list** or **Edit contribution** web parts, the **Check permissions** property of the web part needs to be enabled and the current user must have appropriate permissions to edit the document, as described in Modules -> User contributions -> Security. When using the Groups module versions of the web parts, the icon is displayed either under the same conditions as described above or when the current user is the administrator of the respective group.



8.41.6 Integration with Document library

Files stored in a Document library can also be edited in a client application using WebDAV directly on the live site. There is the **Edit** action available in their drop-down menu, as can be seen in the screenshot below. Clicking the icon opens the document for editing in a client application (if there is an application for the particular file type installed on the client machine and if the application supports WebDAV).

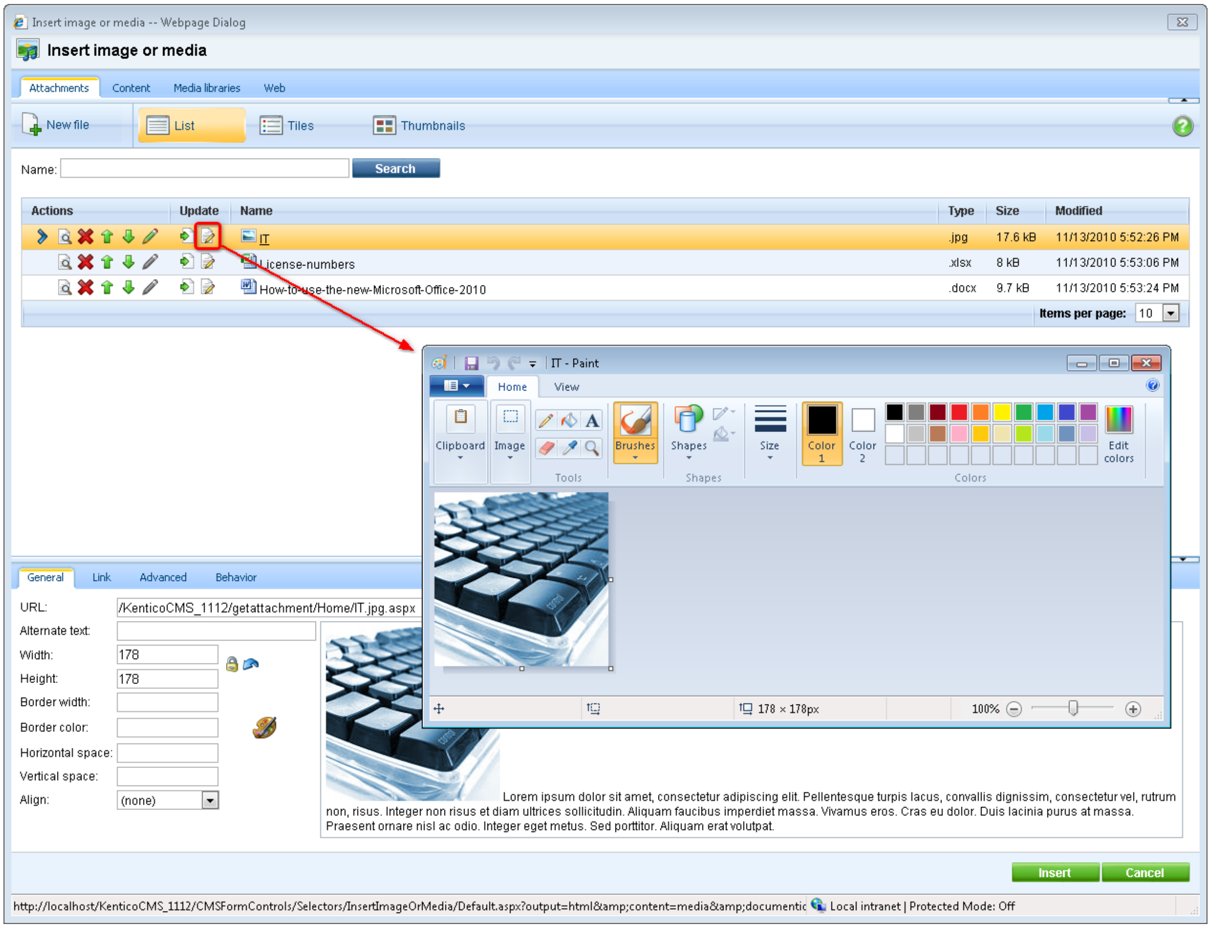
For this action to be enabled, the document must have one of the supported extensions (see Settings for more details) and the **Modify** permission for the particular document must be granted to the current user. Please refer to Modules -> Document library -> Security for more details on permissions in document libraries.



8.41.7 Integration with WYSIWYG editor dialogs

WebDAV editing is also integrated in the Insert/Edit image or media and Insert/Edit link dialogs of the WYSIWYG editor. In these dialogs, the **Edit in client application (** **)** icon is displayed on the **Content** and **Attachments** tabs, next to files with the supported file extensions (see Settings for more details).

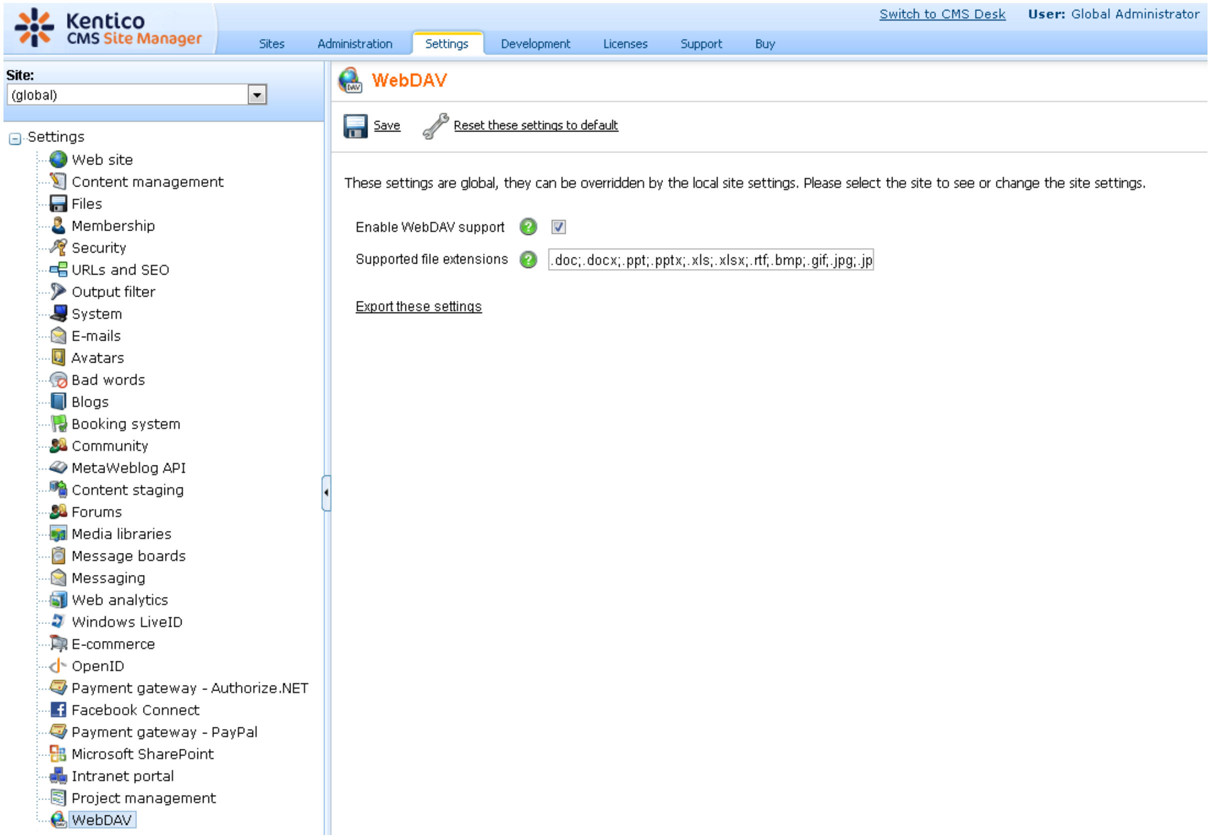
For the icon to be displayed on the **Attachments** tab, the **Modify** permission for the current document needs to be granted to the current user. On the **Content** tab, the icon is displayed only with document for which the **Modify** permission is granted to the current user. Please refer to Development -> Membership -> Permissions -> Document permissions for more details on the three-level document permissions hierarchy in Kentico CMS.



8.41.8 Settings

Settings related to WebDAV integration can be found in **Site Manager -> Settings -> WebDAV**. The following settings are available:

- **Enable WebDAV support** - this options enables/disables WebDAV editing
- **Supported file extensions** - list of file extensions that should be editable using WebDAV (with or without the leading dots, separated by semicolons; e.g. *.docx;.xlsx;.pptx;jpg;jpeg*); appropriate client application with WebDAV support must be installed on the client machine for each listed extension in order for WebDAV editing of these files to be possible



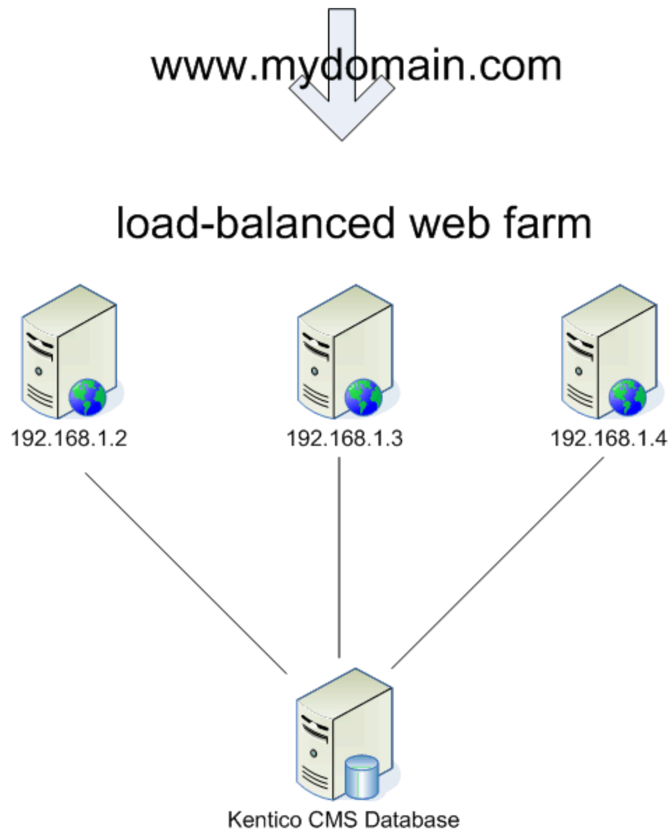
8.42 Web farm synchronization

8.42.1 Overview

Native web farm support in Kentico CMS provides the following features:

- It synchronizes the changes made to the site settings on one of the servers to all the other servers.
- It synchronizes the files uploaded to the site to all other servers. This is used only if you store the uploaded files on the disk or on both disk and in the database.

The following image shows the structure of the web farm and how the synchronization works:



If you change some settings or upload a file using server 192.168.1.2, the other servers do not know about it in a standard scenario. However, if you're using the **Web farm synchronization** module, the module automatically creates a new synchronization task in the database and notifies the other servers once per request so that they process their synchronization task.

To learn how to add web farm servers into the system and configure them, please see the [Defining web farm servers]() topic.

The [Web farm synchronization internals and API]() sub-chapter provides information about the database tables and classes used by the module and examples of how web farm synchronization can be managed using the API.

Using Kentico CMS on a web farm without the Web farm synchronization module

You can use Kentico CMS on a web farm even if you do not use the web farm synchronization module, especially if you do not store uploaded files in the file system. Then, the only limitation is that if you change the settings or page content on one of the servers, the other servers may keep using the old version of the settings in their memory/cache until the web application is restarted or cache content expires.

Please note: The web farm support **doesn't replace any load-balancing or web farm management tools**.

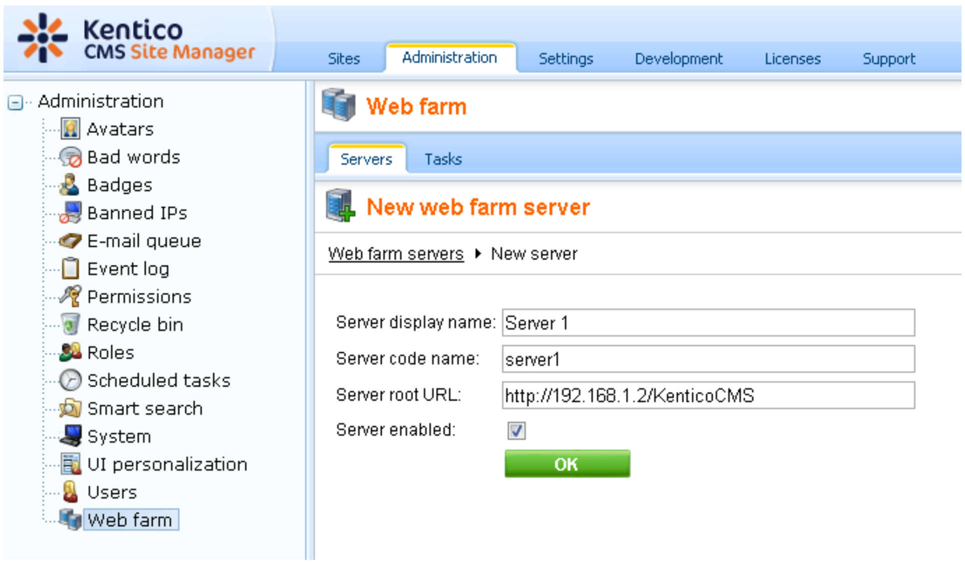
8.42.2 Defining web farm servers

You need to enter the **internal** URLs of all your servers into the system so that the web farm synchronization module knows which servers should be synchronized.

Go to **Site Manager -> Administration -> Web farm** and click **New server**, enter the values:

- **Server display name**: some descriptive name displayed to the users
- **Server code name**: code name of the server that will be used in the server's configuration file
- **Server root URL**: the URL of the root of the website, such as *http://192.168.1.2/KenticoCMS*

Click **OK** to save and repeat the process for every server in your web farm.



Now you need to update the web.config file on each server and add the following keys into the **/configuration/appSettings** section:

```
<add key="CMSWebFarmServerName" value="server1"/>
<add key="CMSWebFarmEnabled" value="true"/>
```

Where server1 is the appropriate code name of the given server. Every server must contain only one such key with its own name.

Additional low-level settings of web farm synchronization can be done by adding the keys listed in the Web farm synchronization settings section of Appendix C - Web.config parameters into the **/configuration/appSettings** section of your *web.config* file.

If the synchronization doesn't work as expected, you can check the failed synchronization tasks on the **Tasks** tab. Processed tasks are automatically removed from the list and failed tasks are displayed with an error message.

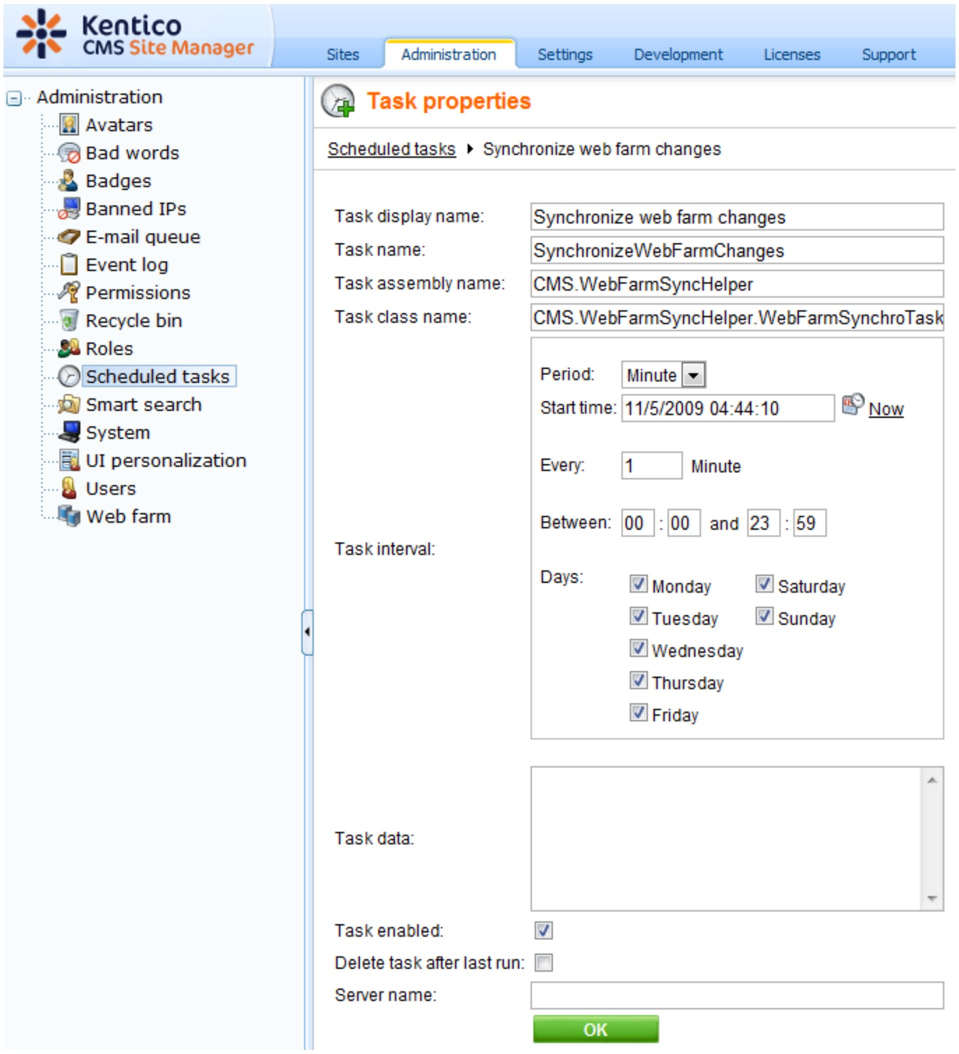
License keys

Please be sure to enter an appropriate license key for the internal URL of the server.

E.g. if the web farm servers are used for domain name *example.com* and you access them internally through URLs like *http://192.168.1.2*, *http://192.168.1.3*, etc., you need to enter separate license keys for *example.com*, *192.168.1.2* and *192.168.1.3* in **Site Manager -> Licenses** dialog (you need to do that on one server only and restart the other instances using **Site Manager -> Administration -> System -> Restart application**).

Configuring the server synchronization interval

By default, web farm servers are updated with changes made on other servers once per request. Synchronizing this frequently may however be impractical for high-traffic websites. The synchronization interval can be changed by setting the value of the **CMSWebFarmUpdateWithinRequest** web.config key to *false* and using the **Synchronize web farm changes** scheduled task instead, which has a configurable execution interval. To do this, go to **Site Manager -> Administration -> Scheduled tasks**, select *(global)* from the **Site** drop-down list at the top of the page, edit () the mentioned task, set the required **Task interval**, check the **Task enabled** property and click **OK**.



The task must also be created for each server in the web farm, so return to the list of scheduled tasks, click the **New task** link, fill in the same properties as those of the **Synchronize web farm changes** task (the **Task name** must be different) and check the **Create tasks for all web farm servers** box below the **Server name** property. Click **OK** to complete the process.

8.42.3 Web farm synchronization internals and API

8.42.3.1 Database tables and API classes

The Web farm synchronization module uses the following database tables:

- **CMS_WebFarmServer** - contains web farm servers and information about them
- **CMS_WebFarmServerTask** - assigns web farm synchronization tasks to the servers that should process them
- **CMS_WebFarmTask -** contains web farm synchronization tasks and information about them

The Web farm synchronization API is provided by the following classes:

- **CMS.SettingsProvider** namespace:

- **WebSyncHelperClass** - this class provides functionality for performing synchronization related tasks

- **CMS.WebFarmSync** namespace:

  - **WebFarmServerInfo**, **WebFarmServerInfoProvider** - these classes provide functionality for managing web farm servers
  - **WebFarmTaskInfo**, **WebFarmTaskInfoProvider** - these classes provide functionality for managing web farm synchronization tasks

- **CMS.WebFarmSyncHelper** namespace

  - **WebSyncHelper** - this class provides further functionality for performing synchronization related tasks

The following chapters show examples of how these classes can be used:

- Getting web farm synchronization task data
- Managing web farm server
- Managing web farm synchronization tasks

The API programming and Kentico CMS internals section of this guide contains more API related information, so please refer to it if required.

For detailed API documentation, such as a list of all methods from the classes above, please refer to Kentico CMS API Reference that is a part of your Kentico CMS installation and can be accessed through the programs folder in Windows Start menu.

8.42.3.2 Getting web farm synchronization task data

The following sample code shows how you can get a web farm synchronization task as a *WebFarmTaskInfo* object:

[C#]

```
using CMS.WebFarmSync;

...

        // Get web farm task object by ID
        WebFarmTaskInfo wfti = WebFarmTaskInfoProvider.GetWebFarmTaskInfo(10);
```

The following sample code shows how you can get a DataSet containing synchronization tasks from the system:

[C#]

```
using System.Data;
using CMS.WebFarmSync;
```

```
...

        string where = "";
        string orderby = "";

        // Get DataSet of web farm tasks by where condition and orderby
        DataSet ds = WebFarmTaskInfoProvider.GetWebFarmTasks(where, orderby);
```

8.42.3.3 Managing web farm servers

The following sample code shows how you can create a new web farm server and add it to the system:

[C#]

```
using CMS.WebFarmSync;

...

        // Create new web farm server object
        WebFarmServerInfo wfsi = new WebFarmServerInfo();

        // Set properties
        wfsi.ServerDisplayName = "Test server";
        wfsi.ServerName = "TestServer";
        wfsi.ServerEnabled = true;
        wfsi.ServerURL = "http://localhost/KenticoCMS";

        // Save to database
        WebFarmServerInfoProvider.SetWebFarmServerInfo(wfsi);
```

The following sample code shows how you can modify the display name of an existing web farm server:

[C#]

```
using CMS.WebFarmSync;

...

        // Get web farm server
        WebFarmServerInfo wfsi = WebFarmServerInfoProvider.GetWebFarmServerInfo("Tes

        // If server exists
        if (wfsi != null)
        {
            // Update property
            wfsi.ServerDisplayName += " updated";

            // Save to database
```

```
                WebFarmServerInfoProvider.SetWebFarmServerInfo(wfsi);
            }
```

The following sample code shows how you can delete an existing web farm server:

[C#]

```
using CMS.WebFarmSync;

...

        // Get web farm server
        WebFarmServerInfo wfsi = WebFarmServerInfoProvider.GetWebFarmServerInfo("Tes

        // If server exists
        if (wfsi != null)
        {
            // Remove from database
            WebFarmServerInfoProvider.DeleteWebFarmServerInfo(wfsi.ServerID);
        }
```

8.42.3.4 Managing web farm synchronization tasks

The following sample code shows how you can create a new web farm synchronization task of a predefined task type and add it to the system:

[C#]

```
using CMS.SettingsProvider;

...

        // Set parameters
        string taskTarget = "";
        string taskTextData = null;
        byte[] taskBinaryData = null;
        WebFarmTaskTypeEnum webfarmTaskType = WebFarmTaskTypeEnum.ClearHashtables;

        // Create web farm task
        WebSyncHelperClass.CreateTask(webfarmTaskType, taskTarget, taskTextData, tas
```

The following sample code shows how you can process an existing web farm synchronization task:

[C#]

```
using CMS.WebFarmSync;
using CMS.WebFarmSyncHelper;
```

```
...

        // Get web farm task
        WebFarmTaskInfo wfti = WebFarmTaskInfoProvider.GetWebFarmTaskInfo(10);

        // If task exists
        if (wfti != null)
        {
            // Process task
            WebSyncHelper.ProcessTask(wfti.TaskGUID);
        }
```

The following sample code shows how you can process all web farm synchronization tasks assigned to the current server (the server that is specified by code name in the web.config file):

[C#]

```
using CMS.WebFarmSyncHelper;

...

        // Process all tasks for server in web.config
        WebSyncHelper.ProcessMyTasks();
```

The following sample code shows how you can delete an existing web farm synchronization task:

[C#]

```
using CMS.WebFarmSync;

...

        // Get web farm task
        WebFarmTaskInfo wfti = WebFarmTaskInfoProvider.GetWebFarmTaskInfo(10);

        // If task exists
        if (wfti != null)
        {
            // Remove from database
            WebFarmTaskInfoProvider.DeleteWebFarmTaskInfo(wfti.TaskID);
        }
```

Part IX

API programming and Kentico CMS internals

9 API programming and Kentico CMS internals

9.1 Overview

Kentico CMS allows you to script any action that you normally make through the user interface, such as:

- content management (select/create/update/delete documents)
- workflow management
- security (create users, roles, set permissions, etc.)
- and many others

This allows you to create customized features and add them to the user interface or write procedures for integration with external systems.

9.2 CMSContext class

CMS.CMSHelper.CMSContext class provides useful methods that allow you to access information about the current page, user, etc. All methods are static, which means you can access them directly like `CMS.CMSHelper.CMSContext.CurrentAliasPath`, without instantiating the CMSContext class.

CurrentAliasPath	Alias path of the currently required document.
CurrentDocument	Currently processed document. This object contains all document data including the product data. Please note that retrieving this property is time consuming because it may need to access the dabatase to get the data. You can use the property CurrentPageInfo to get basic document data without additional operations required. It is cached when Content caching is enabled.
CurrentPageInfo	Currently processed page info. This property provides only basic document fields, such as *NodeID*, *NodeName*, *AliasPath*, *ClassName*, etc. It is cached if Page info caching is enabled.
CurrentSite	Provides information on the current site.
CurrentSiteName	Provides code name of the current site.
CurrentUser	Provides information on the current user and their preferences. Beside the standard UserInfo properties, it also provides the following ones: **PreferredCultureCode** – gets/sets user's preferred content culture. You can use this property to change the culture of the displayed content. **PreferredUICultureCode** – gets/sets user's preferred UI culture. It's used mainly by Kentico CMS webparts, controls and in the administration interface. **IsAuthorizedPerResource** – returns true if the current user is authorized with given permission for given module (resource). **IsAuthorizedPerClassName** – returns true if the current user is authorized with given permission for the given document type (class name). It checks the global document permissions.

	IsAuthorizedPerTreeNode – returns true if the current user is authorized with given permission for given document (node). **IsPublic** – returns true if the current user is the Public user account used for anonymous visitors.
ViewMode	The view mode in which the documents are displayed (edit, design, form, live site, preview).
ResolveCurrentPath	Returns current path or its part according to the provided formatting string. You can use it to get only the first N levels of the path – e.g.: */{0}/{1}* applied on *"/products/computers/ibm"* results in: *"/products/computers"*

9.3 TreeHelper class

CMS.CMSHelper.TreeHelper class provides useful methods that allow you to access documents. All methods are static, which means you can access them directly like `CMS.CMSHelper.TreeHelper.SelectNodes`.

SelectSingleNode	Returns single node (document) specified by given NodeID or path.
SelectSingleDocument	Return single node (document) specified by given DocumentID (culture-specific document version).
SelectNodes	Returns a DataSet object containing nodes (documents) of given path.

9.4 Managing documents

9.4.1 Content management internals

This chapter describes the structure of the content stored in the Kentico CMS repository and how they can be accessed.

Database structure of the documents

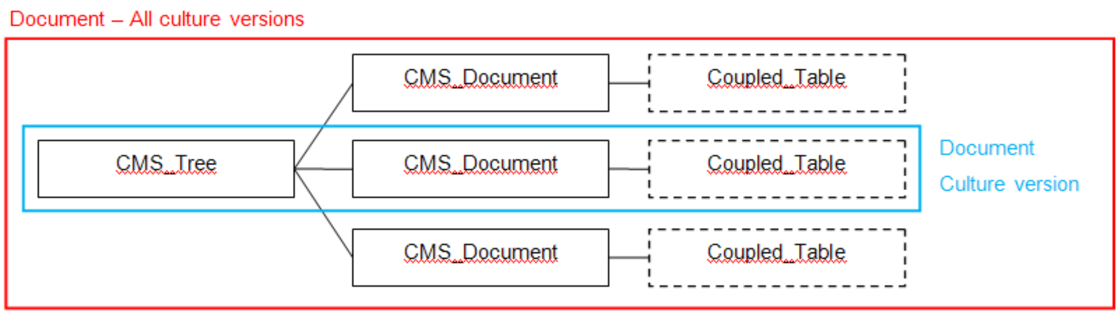
Document data is stored in several joined tables that are used for the tree structure, multilingual support and custom document fields. The document record consists of the following tables:

- **CMS_Tree** – a table with basic document data shared between different language versions of same document. This table determines the tree structure of the website document content. Table contains **one record for all culture versions** of the document. The table does not contain any versioned data.

- **CMS_Document** – a table with document data of specified language version of the document. Table contains **several records** for every document, each one representing one language version of the document. Some of the document columns are versioned columns.

- **Coupled table** – a table that contains document-type specific fields defined by the developer. For example the News document type has a Content_News coupled table that contains NewsTitle, NewsSummary, NewsText and other fields specific for news. Coupled table primary key is referenced by the value of **DocumentForeignKeyValue** column and the table is determined by type of the document. Container document types, such as folder, do not have any coupled table. All columns from coupled tables are versioned. Each culture version of the coupled document contains one record in the coupled table.

The following figure shows how the tables are connected:



See **Kentico CMS Database Reference** or the database structure for details on the table columns.

TreeNode Class

The **CMS.TreeEngine.TreeNode** class encapsulates the tables described above. You can work with the document values using the TreeNode class properties (if available) or using the **SetValue** and **GetValue** methods with column name from either CMS_Tree, CMS_Document or coupled table. There are following methods to make the changes to the database:

- **Insert** – Inserts complete document record to the database, creates new record in CMS_Tree, CMS_Document and coupled table. Use this method for first culture version of the document.
- **InsertAsNewCultureVersion** – Inserts a new language version of the document. Creates new record in CMS_Document and coupled table bound to the existing CMS_Tree record. Use for the new language versions of the same document if some language version of the document already exists.
- **Update** – Updates the document record (all three tables) in the database
- **Delete** – deletes the document record from the database. It deletes the CMS_Document and coupled table record. If the document is the last culture version of the document, it deletes also the CMS_Tree record.

The next chapters contain code examples of working with documents.

See also: Managing attachments, Versioning internals, Workflow internals

9.4.2 Select document(s)

Select single node by NodeID

[C#]

```
using CMS.SiteProvider;

// Tree node
CMS.TreeEngine.TreeNode node = null;
// Tree provider
UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
CMS.TreeEngine.TreeProvider tree = new CMS.TreeEngine.TreeProvider(ui);

// Get Single node specified by it`s ID
node = tree.SelectSingleNode(nodeId);

// Get node name
string result = "The node name is: " + node.NodeName;
```

Reading document properties

Once you retrieve the TreeNode instance representing a document, you can use the GetValue method to retrieve the document properties:

[C#]

```
// Get NewsTitle value of the News document
string newsTitle = (string) node.GetValue("NewsTitle");
```

Reading/setting editable region content

Once you retrieve the TreeNode instance representing a document, you can use the DocumentContent property to retrieve or set the document properties:

[C#]

```
// Get mainText region content of the node representing a page
string mainText = (string) node.DocumentContent["mainText"];

// Set mainText region content of the node representing a page (for portal engine
pages)
node.DocumentContent.EditableWebParts["mainText"] = "my text";

// Set mainText region content of the node representing a page (for ASPX page
templates)
node.DocumentContent.EditableRegions["mainText"] = "my text";
```

Please note: if you need to retrieve these values for the currently displayed page, you can use the **`CMSContext.CurrentPageInfo.EditableItems`** property.

Select single node by AliasPath

[C#]

```
using CMS.SiteProvider;

// Tree node
CMS.TreeEngine.TreeNode node = null;
// Tree provider
UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
CMS.TreeEngine.TreeProvider tree = new CMS.TreeEngine.TreeProvider(ui);

// Get Single node specified by it`s site name, aliaspath And culture code
node = tree.SelectSingleNode("CorporateSite", "/Products/Notebooks/FS-V2030", "en-
us");

string result = "The node name is: " + node.NodeName;
```

Select single document by DocumentID

Please note: the difference between DocumentID and NodeID is that DocumentID is specific for particular language version of the document.

[C#]

```
using CMS.SiteProvider;

// Tree node
CMS.TreeEngine.TreeNode node = null;
// Tree provider
UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
CMS.TreeEngine.TreeProvider tree = new CMS.TreeEngine.TreeProvider(ui);

// Get Single document node specified by it`s ID
node = tree.SelectSingleDocument(documentId);

string result = "The name of the document (node) is: " + node.DocumentName;
```

Select multiple documents

[C#]

```
using CMS.SiteProvider;

DataSet ds = null;

// create a TreeProvider instance
UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
CMS.TreeEngine.TreeProvider tree = new CMS.TreeEngine.TreeProvider(ui);

// get dataset of tree nodes specified by alias path and class names (separated by
semicolon),
```

```
// the second parameter says whether to return default culture documents if the
required
// document language version is not available
ds = tree.SelectNodes("CorporateSite","/Products/%", "en-us", True, "cms.menuitem;
cms.products");

// do something with dataset ...
```

9.4.3 Create document

Creating a new document

[C#]

```
string newsName = "My first news";
// create tree provider instance
CMS.TreeEngine.TreeProvider provider = new CMS.TreeEngine.TreeProvider(CMS.
CMSHelper.CMSContext.CurrentUser);

// get parent node for new document
CMS.TreeEngine.TreeNode parent = provider.SelectSingleNode(CMS.CMSHelper.
CMSContext.CurrentSiteName, "/News", "en-us");

// create a new tree node
CMS.TreeEngine.TreeNode node = new CMS.TreeEngine.TreeNode("cms.news", provider);

if (parent != null)
{
    // set document properties
    node.NodeName = newsName;
    node.NodeAlias = newsName;
    node.SetValue("DocumentCulture", "en-us");
    node.SetValue("NewsTitle", newsName);
    node.SetValue("NewsSummary", "Some summary.");
    node.SetValue("NewsText", "Some text.");
    node.SetValue("NewsReleaseDate", DateTime.Now);
    // create New document
    node.Insert(parent.NodeID);
}
```

9.4.4 Update document

Update an existing document

[C#]

```
// tree provider
CMS.TreeEngine.TreeProvider provider = new CMS.TreeEngine.TreeProvider(CMS.
CMSHelper.CMSContext.CurrentUser);
// get document with specified site, alias path and culture
```

```
CMS.TreeEngine.TreeNode node = provider.SelectSingleNode(CMS.CMSHelper.CMSContext.
CurrentSiteName, "/News/MyFirstNews", "en-us");

if (node != null)
{
   // change some values
   node.SetValue("NewsSummary", "Some changed summary.");
   node.SetValue("NewsText", "Some changed text.");
   // Update the document
   node.Update();
}
```

9.4.5 Delete document

Deleting a document

[C#]

```
// Tree provider
CMS.TreeEngine.TreeProvider provider = new CMS.TreeEngine.TreeProvider(CMS.
CMSHelper.CMSContext.CurrentUser);

// Get document with specified site, aliaspath and culture
CMS.TreeEngine.TreeNode node = provider.SelectSingleNode(CMS.CMSHelper.CMSContext.
CurrentSiteName, "/News/MyFirstNews", "en-us");

if (node != null)
{
   // Delete the document
   node.Delete();
}
```

9.4.6 Copy document

Copying a document

[C#]

```
// Tree provider
CMS.TreeEngine.TreeProvider provider = new CMS.TreeEngine.TreeProvider(CMS.
CMSHelper.CMSContext.CurrentUser);
// Get document (node) which will be copied
CMS.TreeEngine.TreeNode node = provider.SelectSingleNode(CMS.CMSHelper.CMSContext.
CurrentSiteName, "/Products/PDAs/ItemA", "en-us");
// Get target node under which the document will be copied
CMS.TreeEngine.TreeNode target = provider.SelectSingleNode(CMS.CMSHelper.
CMSContext.CurrentSiteName, "/Products/Notebooks", "en-us");

if ((target != null) && (node != null))
{
```

```
   // copy the document under the specified target,
   // the third parameter says that document`s child nodes (if any) will not be
copied
   provider.CopyNode(node, target.NodeID, false);
}
```

9.4.7 Move document

Moving a document

[C#]

```
// Tree provider
CMS.TreeEngine.TreeProvider provider = new CMS.TreeEngine.TreeProvider(CMS.
CMSHelper.CMSContext.CurrentUser);

// Get document (node) which will be moved
CMS.TreeEngine.TreeNode node = provider.SelectSingleNode(CMS.CMSHelper.CMSContext.
CurrentSiteName, "/Products/PDAs/ItemA", "en-us");

// Get target node under which the document will be moved
CMS.TreeEngine.TreeNode target = provider.SelectSingleNode(CMS.CMSHelper.
CMSContext.CurrentSiteName, "/Products/Notebooks", "en-us");

if ((target != null) && (node != null))
{
   // Move the document under the specified target
   provider.MoveNode(node, target.NodeID);
}
```

9.4.8 Using transactions when managing documents

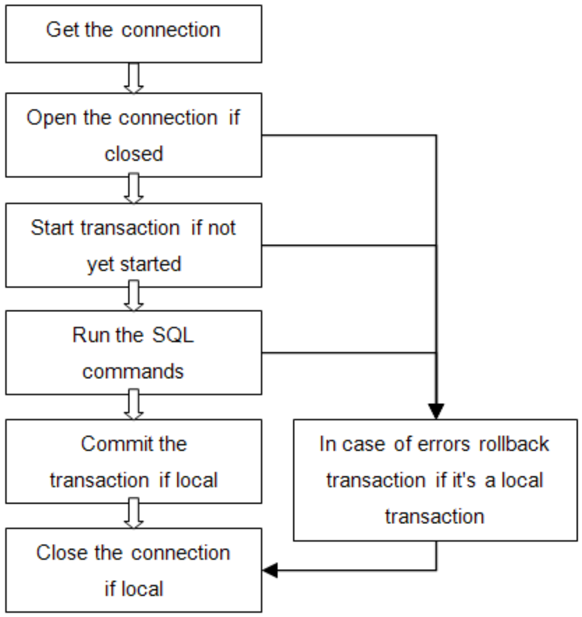
This chapter explains how to work with transactions in Kentico CMS API. Transactions are usually used to ensure database consistency.

Native transactions

Kentico CMS uses SQL transactions to ensure the system database consistency when performing complex operations consisting of multiple steps. In such cases the whole operation forms a single block of SQL commands that can be rolled back when an error occurs or committed if the complete operation was successful. Transaction handling is provided by **CMS.DataEngine** library, in the **GeneralConnection** class.

Please note that you should always use single connection for every operation within the open transaction to avoid deadlocks.

The transactions flow in Kentico CMS looks like this:



Please note: Always pass the existing connection as a method parameter if you call the method that uses DB access within transaction.

Example

Please use following example as a template in all the methods that work with transactions:

[C#]

```
using CMS.DataEngine;
using CMS.TreeEngine;
using CMS.FileManager;
using CMS.WorkflowEngine;

/// <summary>
/// Sample method
/// </summary>
/// <param name="conn">Data connection to use</param>

public void DoSomethingInTransaction(GeneralConnection conn)
{

    // Process within transaction
    using (var tr = new CMSTransactionScope(conn))
    {
        // --- HERE YOU CAN USE YOUR DATABASE ACCESS CODE LIKE: ----
        DataSet ds = conn.ExecuteQuery("cms.user.selectall", null, null, null);

        // --- IF YOU NEED TO USE TREEPROVIDER, ALWAYS INSTATIATE IT WITH EXISTING
CONNECTION
        TreeProvider tree = new TreeProvider(null, conn);

        // --- ALWAYS USE METHOD OVERRIDES THAT USE CONNECTION (TREEPROVIDER)
```

```
PARAMETER ---
        AttachmentInfo ai = DocumentHelper.GetAttachment(Guid.NewGuid(), 1, tree);

        // Commit transaction
        tr.Commit();
    }

}
```

You may also work with **TreeProvider** class instead of connection object. In that case, use its **Connection** property the same way like the **GeneralConnection** in this example and use the same connection object in all operations inside the transaction.

9.4.9 Create linked document

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;

...

            // Prepare the TreeProvider
            // it must be initialized with user information when editing document
structure
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);

            // Get the document (current culture)
            CMS.TreeEngine.TreeNode node = tree.SelectSingleNode(CMSContext.
CurrentSiteName, "/TestingNews", CMSContext.CurrentUser.PreferredCultureCode,
false, null, false);
            if (node != null)
            {
                // Update the data
                node.SetValue("NodeAlias", "TestingNewsLink");

                // Insert new culture version of the document in the same site
section
                node.InsertAsLink(node.NodeParentID);

                this.lblInfo.Text = "New link '" + node.NodeAliasPath + "' to the
document '/TestingNews' has been created.";
            }
```

9.4.10 Delete linked document

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...

            // Prepare the TreeProvider
            // it must be initialized with user information when editing document
structure
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);

            // Get the linked document by alias path (any culture)
            CMS.TreeEngine.TreeNode node = tree.SelectSingleNode(CMSContext.
CurrentSiteName, "/TestingNewsLink", TreeProvider.ALL_CULTURES, false, null, false
);
            if (node != null)
            {
                // Always delete the document with DocumentHelper
                // it handles all the dependencies
                DocumentHelper.DeleteDocument(node, tree, true, true, true);

                this.lblInfo.Text = "Link '" + node.NodeAliasPath + "' has been
deleted.";
            }
```

9.5 Managing attachments and files (cms.file)

9.5.1 Managing attachments

The documents may contain a custom field of type "file". The field itself does not contain attachment data. Only **AttachmentGUID** (unique identifier) value is stored in the document field referencing the record within separate table.

The attachments are stored in the table **CMS_Attachment** which contains the attachment metadata (file name, size, type, etc.) and optionally the attachment binary data (depending on where the files are stored). Attachment file may also be stored in the filesystem if the system is configured for storing files in the database. You can work with the attachments using the **AttachmentManager** class methods.

Here are some examples how to work with attachments:

Adding a new attachment to a document

To add a new attachment to a document, you need to save the attachment and store its GUID (unique identifier) to the document field.

[C#]

```
using CMS.SiteProvider;
using CMS.TreeEngine;
using CMS.CMSHelper;
using CMS.FileManager;

// Always work with some user when editing documents
UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
TreeProvider tree = new TreeProvider(ui);

// First, get the document
CMS.TreeEngine.TreeNode node =
tree.SelectSingleNode(CMSContext.CurrentSite.SiteName, "/Files/home", "en-us");
if ( node != null )
{
  // Create the attachment manager
  AttachmentManager am = new AttachmentManager(tree.Connection);

  // Create the attachment
  Guid attachmentGuid = Guid.NewGuid();
  AttachmentInfo ai =
    new AttachmentInfo("c:\\home.gif", node.DocumentID, attachmentGuid, tree.
Connection);
  am.SetAttachmentInfo(ai);

  // Set the attachment reference
  node.SetValue("FileAttachment", attachmentGuid);
  node.Update();
}
```

Removing an attachment from a document

To remove an existing attachment from a document, you need to delete the attachment and remove the document field reference to it. You should always remove the old attachment if you insert a new one.

[C#]

```
using CMS.GlobalHelper;
using CMS.FileManager;
using CMS.CMSHelper;

// ... this code continues from the previous example

// Get current attachment GUID
Guid existingGuid = ValidationHelper.GetGuid(node.GetValue("FileAttachment"), Guid
.Empty);
if (existingGuid != Guid.Empty)
{
  // Get the attachment
  AttachmentInfo existingAttachment =
    am.GetAttachmentInfo(existingGuid, CMSContext.CurrentSite.SiteName);
  if ( existingAttachment != null )
  {
```

```
    // Delete the attachment
    am.DeleteAttachmentInfo(existingAttachment.AttachmentID);
  }

  // Clear the attachment reference from the document field
  node.SetValue("FileAttachment", null); ;
  node.Update();
}
```

Please note: if you use workflow and versioning, see Versioning internals and the related chapters for more details how to work with attachments in the versioned documents.

9.5.2 Managing files (cms.file)

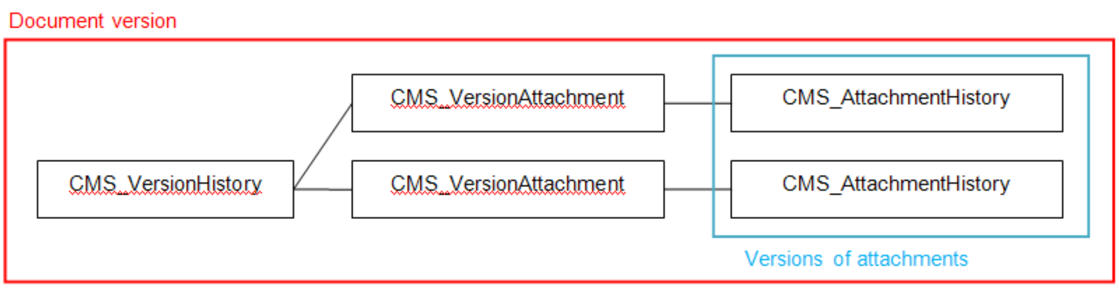
The File document type (cms.file) is actually just another document type and you can modify it using the same methods that you use for other document types. The actual file (uploaded image, Word document or other file type) is stored as an attachment in field **FileAttachment**.

9.6 Managing documents with workflow

9.6.1 Versioning internals

For versioned documents (with workflow assigned), only the published data is stored in the tables described in chapter Content management internals. Document versions and its history are stored in separate tables:

- **CMS_VersionHistory** – Table containing the document versions. Each version of the document is represented by one record that contains the complete document data in XML.
- **CMS_AttachmentHistory** – Table containing the attachment version records. Each attachment version is bound to one or more document versions. Every record contains binary data of the attachment.
- **CMS_VersionAttachment** – Table containing the bindings of the attachment versions to the document versions.



When the document is **published**, version history record is taken and the document data are saved to the regular database structure (CMS_Tree, CMS_Document, coupled table) and current attachments are replaced by the attachment versions corresponding to the published document version.

Working with the versioned documents

Since the **TreeNode** methods can work only with the published records, there is an interface provided to work with the versioned documents. This interface is provided by classes **DocumentHelper**, **WorkflowManager** and **VersionManager** in the **CMS.WorkflowEngine** library. You should always use **DocumentHelper** for the document operations if not sure whether the document is versioned or not, the DocumentHelper methods will choose the proper actions for you. Basic methods to work with documents are:

- **InsertDocument** – Inserts a new document; versioned equivalent to **TreeNode.Insert** method.
- **InsertNewCultureVersion** – Inserts a new culture version of the document; versioned equivalent to **TreeNode.InsertAsNewCultureVersion** method.
- **UpdateDocument** – Updates a document; versioned equivalent of the **TreeNode.Update** method. If the document is versioned, it updates the latest (checked out) version of the document.
- **DeleteDocument** – Deletes a document. Versioned equivalent to the **TreeNode.Delete** which also deletes the depending records from the version history and attachment tables. You should always use this method to delete a document which is (was) versioned or contains any attachment fields.
- **GetDocument(s)** – Gets the latest version of the document(s) from the version history record(s). Equivalent to **TreeProvider.SelectSingleNode** and **TreeProvider.SelectNodes**.

9.6.2 Workflow internals

The workflow process is managed by the methods of the **WorkflowManager** class. Document uses workflow if there is a workflow scope defined for the document type and document location (SiteManager –> Development –> Workflow). If the document is in the middle of the workflow process and the workflow has been removed from the given group of documents, the current workflow must be finished (document must be published) in order to disable the workflow for the document.

There are three default workflow steps that are always used in the workflow process:

- **Edit** – A new version of the document is edited.
- **Published** – Document version was approved to be published and is either published or waiting to be published (if it's a scheduled document).
- **Archived** – Document is archived, it is no longer published but it's stored for archive purposes.

You add **your own workflow steps** between the **Edit** and **Published** steps.

Here are some examples on how to use the workflow API:

Getting the document

You always need to get the proper document version first in order to keep the document data consistent if working with the versioned data. You can use both published and latest document version of the document for the basic workflow operations since it affects only the not-versioned data.

[C#]

```
using CMS.SiteProvider;
using CMS.TreeEngine;
using CMS.CMSHelper;
using CMS.WorkflowEngine;

// Always work with some user when editing documents
UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
TreeProvider tree = new TreeProvider(ui);

// First, get the document
CMS.TreeEngine.TreeNode node =
tree.SelectSingleNode(CMSContext.CurrentSite.SiteName, "/News/Your-first-news",
"en-us");
```

Getting the document workflow information

You can use the **WorkflowManager.GetNodeWorkflow** method to get the workflow information for the document node.

[C#]

```
using CMS.WorkflowEngine;

// Create the workflow manager instance
WorkflowManager wm = new WorkflowManager(tree);
WorkflowInfo wi = wm.GetNodeWorkflow(node);
if (wi != null)
{
  // Document is using workflow
}
else
{
  // Not using workflow
}
```

You can also use the **WorkflowManager.GetNodeWorkflowScope** method to check whether there is some workflow scope defined for the given document.

Getting current workflow step

The current workflow step ID is stored in the document field **DocumentWorfklowStepID**. Please note: if the workflow is defined and step ID is missing, the document is in the “edit” step.

[C#]

```
using CMS.WorkflowEngine;

if (wi != null)
{
  // Document is using workflow, get current step information
```

```
  WorkflowStepInfo si = wm.GetStepInfo(node);
  if (si != null)
  {
    // Use the workflow step information
  }
}
```

Changing the workflow step

You should always follow the workflow definition and use the built-in methods only to move between the workflow steps to ensure the system consistency. Workflow process moves one step at a time, you can move through several steps by calling the methods several times. You should not attempt to move before the “edit” step and after the “archived” step. Use **WorkflowManager.MoveToNextStep** (the "approve" action) and **WorkflowManager.MoveToPreviousStep** (the "reject" action).

[C#]

```
if (si != null)
{
  // Use the workflow step information to approve the document until it is
published
  while (si.StepName.ToLower() != "published")
  {
    si = wm.MoveToNextStep(node, "");
    if (si == null)
    {
       break;
    }
  }
}
```

Or

[C#]

```
if (si != null)
{
  // Use the workflow step information to reject the document until it is in the
edit step
  while (si.StepName.ToLower() != "edit")
  {
    si = wm.MoveToPreviousStep(node, "");
    if (si == null)
    {
       break;
    }
  }
}
```

To move the document to the **Archived** step, use **WorkflowManager.ArchiveDocument** method.

9.6.3 Updating a versioned document

This example shows how to update a versioned document (document with workflow). Workflow and the versioning are very closely connected, there should be always a new document version created if you edit a document that is under workflow.

The basic version management operations are provided by the class DocumentHelper in the methods mentioned in the previous chapter (InsertDocument, …). In order to manage the document versions, use VersionManager class methods.

Here are some examples how to work with versioned documents:

Getting the document

You always need to work with the latest document version, use DocumentHelper.GetDocument instead of just simple TreeProvider.SelectSingleNode.

[C#]

```
using CMS.SiteProvider;
using CMS.TreeEngine;
using CMS.CMSHelper;
using CMS.WorkflowEngine;

// Always work with some user when editing documents
UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
TreeProvider tree = new TreeProvider(ui);

// First, get the base document
CMS.TreeEngine.TreeNode node =
tree.SelectSingleNode(CMSContext.CurrentSite.SiteName, "/News/Your-first-news",
"en-us");
if ( node != null )
{
  // Get latest version of the document
  node = DocumentHelper.GetDocument(node, tree);
}
```

Checking out the document

If you edit the document, you should check it out to create a new document version and lock the document to avoid editing by someone else. Use VersionManager.CheckOut method to check out the document.

[C#]

```
using CMS.WorkflowEngine;

// Check out the document
VersionManager vm = new VersionManager(tree);
vm.CheckOut(node);
```

Updating current document version

Use **DocumentHelper.UpdateDocument** method to update current document version.

[C#]

```
using CMS.WorkflowEngine;

// Update the document
node.SetValue("DocumentName", "New name");
DocumentHelper.UpdateDocument(node, tree);
```

Check in the document

After you edit the document, you should check it in to allow other users to edit it or allow the workflow process to continue. Use VersionManager.CheckIn to do so.

[C#]

```
// Check in the document
vm.CheckIn(node, null, null);
```

9.6.4 Managing workflow schema

The following code samples show how to create a new schema Testing workflow with following steps:

Edit -> **Article Approval (custom step)** -> Publish -> Archive

It also creates a workflow scope that applies this workflow to all articles in the /TestingWorkflow section.

Creating a new workflow schema

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...
            // Prepare the data
            WorkflowInfo wi = new WorkflowInfo();
            wi.WorkflowDisplayName = "Testing workflow";
            wi.WorkflowName = "TestingWorkflow";

            // Insert the workflow
            WorkflowInfoProvider.SetWorkflowInfo(wi);
```

```
            // Create default steps (required for proper operation)
            WorkflowInfoProvider.CreateDefaultWorkflowSteps(wi.WorkflowID);
```

Creating a workflow step

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...

            // Get the workflow
            WorkflowInfo wi = WorkflowInfoProvider.GetWorkflowInfo(
"TestingWorkflow");
            if (wi != null)
            {
                // Init new step
                WorkflowStepInfo wsi = new WorkflowStepInfo();
                wsi.StepDisplayName = "Article approval";
                wsi.StepName = "ArticleApproval";

                // Get published step info for the proper position
                WorkflowStepInfo psi = WorkflowInfoProvider.GetPublishedStep(wi.
WorkflowID);
                if (psi != null)
                {
                    // New step has the previous published step order
                    wsi.StepOrder = psi.StepOrder;

                    // Move the published step down
                    psi.StepOrder += 1;
                    WorkflowInfoProvider.SetWorkflowStepInfo(psi);

                    // Move the archived step down
                    WorkflowStepInfo asi = WorkflowInfoProvider.GetArchivedStep
(wi.WorkflowID);
                    if (asi != null)
                    {
                        asi.StepOrder += 1;
                        WorkflowInfoProvider.SetWorkflowStepInfo(asi);
                    }
                }

                // Insert the step to the database
                wsi.StepWorkflowID = wi.WorkflowID;
                WorkflowInfoProvider.SetWorkflowStepInfo(wsi);
            }
```

Defining a workflow scope

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...

            // Get the workflow
            WorkflowInfo wi = WorkflowInfoProvider.GetWorkflowInfo(
"TestingWorkflow");
            if (wi != null)
            {
                // Init new workflow scope
                WorkflowScopeInfo wsi = new WorkflowScopeInfo();
                wsi.ScopeStartingPath = "/TestingWorkflow";

                // Assign the document type article
                DataClassInfo ci = DataClassInfoProvider.GetDataClass("CMS.
Article");
                if ( ci != null )
                {
                    wsi.ScopeClassID = ci.ClassID;
                }

                wsi.ScopeSiteID = CMSContext.CurrentSite.SiteID;
                wsi.ScopeWorkflowID = wi.WorkflowID;

                // Save the scope to the database
                WorkflowInfoProvider.SetWorkflowScopeInfo(wsi);
            }
```

Deleting a workflow

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;
...

            // Get the workflow
            WorkflowInfo wi = WorkflowInfoProvider.GetWorkflowInfo(
```

```
"TestingWorkflow");

            // Delete the worfklow
            WorkflowInfoProvider.DeleteWorkflowInfo(wi.WorkflowID);
```

9.6.5 A complete example

The following samples show you how to create, approve, publish, reject, archive and delete a document. It uses the sample workflow created in chapter Managing workflow schema:

Edit -> **Article Approval (custom step)** -> Publish -> Archive

Creating a new document

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...

            // Prepare the TreeProvider (it must be initialized with user
information
            // when editing content)
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);

            // Get the parent document
            CMS.TreeEngine.TreeNode parentNode = tree.SelectSingleNode(CMSContext.
CurrentSiteName, "/",                TreeProvider.ALL_CULTURES, true, null, false);
            if (parentNode != null)
            {
                // Create new document
                CMS.TreeEngine.TreeNode node = new CMS.TreeEngine.TreeNode("CMS.
Article", tree);
                node.DocumentName = "TestingWorkflow";
                node.DocumentCulture = CMSContext.CurrentUser.
PreferredCultureCode;

                // Set document fields
                node.SetValue("ArticleName", "Testing workflow");
                node.SetValue("ArticleTeaserText", "Testing article for workflow"
);
                node.SetValue("ArticleText", "Testing article for workflow text");

                // Although article contains the attachment field, the attachment
can be added only                     // after the document has been created (see
below)
```

```
                // Insert the document - DocumentHelper should be used when using
workflow
                DocumentHelper.InsertDocument(node, parentNode.NodeID, tree);

                // Insert the attachment (image) to the document
                if (FileUpload.PostedFile != null)
                {
                    DocumentHelper.AddAttachment(node, "ArticleTeaserImage",
FileUpload.PostedFile,
                      tree);

                    // Update the document
                    DocumentHelper.UpdateDocument(node, tree);
                }
            }
```

Editing a document

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...

            // Prepare the TreeProvider (it must be initialized with user
information when editing
            // document structure)
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);

            // Get the document (current culture)
            CMS.TreeEngine.TreeNode node = tree.SelectSingleNode(CMSContext.
CurrentSiteName,
               "/TestingWorkflow", CMSContext.CurrentUser.PreferredCultureCode,
false, null, false);
            if (node != null)
            {
                // Get the latest version of the document (always use
DocumentHelper when using
                // workflow)
                node = DocumentHelper.GetDocument(node, tree);

                // Check out the document
                VersionManager vm = new VersionManager(tree);
                vm.CheckOut(node);

                // Update the document fields the regular way
                node.SetValue("ArticleName", "Edited testing workflow");
                node.SetValue("ArticleTeaserText", "Edited testing article for
```

```
workflow");
                node.SetValue("ArticleText", "Edited testing article for workflow
text");

                // Update the attachment if present
                if (FileUpload.PostedFile != null)
                {
                    // Add the attachment
                    DocumentHelper.AddAttachment(node, "ArticleTeaserImage",
FileUpload.PostedFile,
                      tree);
                }

                // Update the document
                DocumentHelper.UpdateDocument(node, tree);

                // Check in the document
                vm.CheckIn(node, null, null);
            }
```

Approving the document

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...

            // Prepare the TreeProvider (it must be initialized with user
information when editing
            // document structure)
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);

            // Get the document (current culture)
            CMS.TreeEngine.TreeNode node =  tree.SelectSingleNode(CMSContext.
CurrentSiteName,
               "/TestingWorkflow", CMSContext.CurrentUser.PreferredCultureCode,
false, null, false);
            if (node != null)
            {
                // Get the latest version of the document (always get the document
with DocumentHelper
                // when using workflow)
                node = DocumentHelper.GetDocument(node, tree);

                // Move the document to the next step (approve)
                WorkflowManager wm = new WorkflowManager(tree);
                wm.MoveToNextStep(node, "");
```

```
            }
```

Rejecting the document

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...

            // Prepare the TreeProvider (it must be initialized with user
information when editing
            // document structure)
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);

            // Get the document (current culture)
            CMS.TreeEngine.TreeNode node =  tree.SelectSingleNode(CMSContext.
CurrentSiteName,
               "/TestingWorkflow", CMSContext.CurrentUser.PreferredCultureCode,
false, null, false);
            if (node != null)
            {
                // Get the latest version of the document (always get the document
with DocumentHelper
                // when using workflow)
                node = DocumentHelper.GetDocument(node, tree);

                // Move the document to the next step (approve)
                WorkflowManager wm = new WorkflowManager(tree);
                wm.MoveToPreviousStep(node, "");
            }
```

Publish the document

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...
```

```
            // Prepare the TreeProvider (it must be initialized with user
information when editing
            // document structure)
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);

            // Get the document (current culture)
            CMS.TreeEngine.TreeNode node =  tree.SelectSingleNode(CMSContext.
CurrentSiteName,
               "/TestingWorkflow", CMSContext.CurrentUser.PreferredCultureCode,
false, null, false);
            if (node != null)
            {
                // Get the latest version of the document (always get the document
with DocumentHelper
                // when using workflow)
                node = DocumentHelper.GetDocument(node, tree);

                // Move the document to the next step (approve)
                WorkflowManager wm = new WorkflowManager(tree);

                // Approve until the step is publish
                WorkflowStepInfo currentStep = wm.GetStepInfo(node);
                while ((currentStep != null) && (currentStep.StepName.ToLower() !
= "published"))
                {
                    currentStep = wm.MoveToNextStep(node, "");
                }
             }
```

Archive the document

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...

            // Prepare the TreeProvider (it must be initialized with user
information when editing
            // document structure)
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);

            // Get the document (current culture)
            CMS.TreeEngine.TreeNode node =  tree.SelectSingleNode(CMSContext.
CurrentSiteName,
               "/TestingWorkflow", CMSContext.CurrentUser.PreferredCultureCode,
```

```
false, null, false);
            if (node != null)
            {
                // Get the latest version of the document (always get the document
with DocumentHelper
                // when using workflow)
                node = DocumentHelper.GetDocument(node, tree);

                // Archive the document
                WorkflowManager wm = new WorkflowManager(tree);
                wm.ArchiveDocument(node, "");
            }
```

Delete (destroy) the document

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.TreeEngine;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...

            // Prepare the TreeProvider (it must be initialized with user
information when editing
            // document structure)
            UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
            TreeProvider tree = new TreeProvider(ui);

            // Get the document (current culture)
            CMS.TreeEngine.TreeNode node = tree.SelectSingleNode(CMSContext.
CurrentSiteName,
               "/TestingWorkflow", CMSContext.CurrentUser.PreferredCultureCode,
false, null, false);

            if (node != null)
            {
                // Always delete the document with DocumentHelper, it handles all
the dependencies
                DocumentHelper.DeleteDocument(node, tree, true, true, true);
            }
```

9.7 Security management

9.7.1 Security Internals Overview

The security information is stored in the following tables:

- CMS_User - user information. If you need to import users from an existing database, you can import them directly into this table, there are no special dependencies.

- CMS_Role - roles
- CMS_UserRole - assigns users to roles
- CMS_UserSite - assigns users to sites
- CMS_Resource - contains modules
- CMS_ResourceSite - assigns modules to sites
- CMS_Permission - permissions (permission names, such as "read", "modify", etc.) used for modules (CMS_Resource) and document types (CMS_Class)
- CMS_RolePermission - assigns permissions to user roles
- CMS_Class - contains definitions of data classes and document types

Most of these settings can be managed by the classes available in the CMS.SiteProvider namespace. The following chapters contain API examples of some common actions.

9.7.2 Managing users

Creating a new user

[C#]

```
using CMS.SiteProvider;

// Create UserInfo
UserInfo user = new UserInfo();

// Set some properties
user.UserName = "Alice";
user.FirstName = "Alice";
user.LastName = "Cooper";
user.FullName = "Alice Cooper";
user.Email = "alice.cooper@domain.com";
user.IsEditor = true;
user.PreferredCultureCode = "en-us";

// Create new user
CMS.SiteProvider.UserInfoProvider.SetUserInfo(user);
```

Selecting and updating a user

[C#]

```
using CMS.SiteProvider;

// Get UserInfo of the specified name
UserInfo user = CMS.SiteProvider.UserInfoProvider.GetUserInfo("Alice");

if (user != null)
{
   // Change some values
   user.Enabled = false;
```

```
    // Update the UserInfo
    CMS.SiteProvider.UserInfoProvider.SetUserInfo(user);
}
```

Deleting a user

[C#]

```
CMS.SiteProvider.UserInfoProvider.DeleteUser("Alice");
```

Getting the list of sites the user belongs to

[C#]

```
using CMS.SiteProvider;

// Get UserInfo of the specified name
UserInfo user = CMS.SiteProvider.UserInfoProvider.GetUserInfo("Alice");

if (user != null)
{
    // Get table with user's sites
    DataTable table = CMS.SiteProvider.UserInfoProvider.GetUserSites(user.UserID);

    if (table != null)
    {
       // Process table ...
    }
}
```

Adding user to a site

[C#]

```
// Use UserName and SiteName to add user to site
CMS.SiteProvider.UserInfoProvider.AddUserToSite("Alice", CMS.CMSHelper.CMSContext.
CurrentSite.SiteName);
```

Removing user from a site

[C#]

```
// User UserName and SiteName to remove user from site
CMS.SiteProvider.UserInfoProvider.RemoveUserFromSite("Alice", CMS.CMSHelper.
CMSContext.CurrentSite.SiteName);
```

9.7.3 Setting user password

[C#]

```
using CMS.SiteProvider;

...

UserInfoProvider.SetPassword("testinguser", "newpassword");
```

9.7.4 Managing roles

Creating a new role

[C#]

```
using CMS.SiteProvider;

// Create new RoleInfo
RoleInfo role = new RoleInfo();

// Set RoleInfo properties except RoleID
role.RoleName = "DocumentEditor";
role.DisplayName = "Document editor";
role.Description = "some description...";
role.SiteID = CMS.CMSHelper.CMSContext.CurrentSiteID;

// check if the role name is unique in the given site
if (!CMS.SiteProvider.RoleInfoProvider.RoleExists(role.RoleName, CMS.CMSHelper.
CMSContext.CurrentSiteName))
{
   // Insert new RoleInfo for current site
   CMS.SiteProvider.RoleInfoProvider.SetRoleInfo(role);
}
else
{
   // Role with the same name already exists in specified site
}
```

Selecting and updating a role

[C#]

```
using CMS.SiteProvider;

// Get role of specified name from current site
RoleInfo role = CMS.SiteProvider.RoleInfoProvider.GetRoleInfo("ArticleEditor",
CMS.CMSHelper.CMSContext.CurrentSiteName);

if (role != null)
{
```

```
    // Make some changes
    role.Description = "Changed description.";
    // Update the role
    CMS.SiteProvider.RoleInfoProvider.SetRoleInfo(role);
}
```

Deleting a role

[C#]

```
// Delete the role
CMS.SiteProvider.RoleInfoProvider.DeleteRole("ArticleEditor", CMS.CMSHelper.
CMSContext.CurrentSiteName);
```

9.7.5 Managing user roles

Getting users who belong to the given role

[C#]

```
using CMS.SiteProvider;

// Get role of specified name from current site
RoleInfo role = CMS.SiteProvider.RoleInfoProvider.GetRoleInfo("ArticleEditor",
CMS.CMSHelper.CMSContext.CurrentSiteName);

if (role != null)
{
    // Get datatable of the role members
    System.Data.DataTable users = CMS.SiteProvider.RoleInfoProvider.GetRoleUsers
(role.RoleID);

    if (users != null)
    {
        // the table now contains the users that belong to the given role ...
    }
}
```

Get user's roles

[C#]

```
using CMS.SiteProvider;

// Get UserInfo of the specified name
UserInfo user = CMS.SiteProvider.UserInfoProvider.GetUserInfo("Alice");

if (user != null)
{
```

```
    // Get table with all user roles
    System.Data.DataTable table = CMS.SiteProvider.UserInfoProvider.GetUserRoles
(user);

    if (table != null)
    {
        // the table now contains the roles the user is member of...
    }
}
```

Adding user to role

[C#]

```
// Use UserName, RoleName and SiteName to add user to role
CMS.SiteProvider.UserInfoProvider.AddUserToRole("Alice", "DocumentEditor", CMS.
CMSHelper.CMSContext.CurrentSiteName);
```

Removing user from role

[C#]

```
// Use UserName, RoleName and SiteName to remove user from role
CMS.SiteProvider.UserInfoProvider.RemoveUserFromRole("Alice", "DocumentEditor",
CMS.CMSHelper.CMSContext.CurrentSiteName);
```

Checking if user is member of a given role

See Checking user permissions for more details.

9.7.6 Authenticating user

Checking user name and password

[C#]

```
using CMS.SiteProvider;

// UserInfo
UserInfo user = null;

user = CMS.SiteProvider.UserInfoProvider.AuthenticateUser("Alice", "*******", CMS.
CMSHelper.CMSContext.CurrentSiteName);

if (user != null)
{
   // Authentication was successful
}
```

9.7.7 Checking user permissions

Getting current user's name

[C#]

```
string currentUserName = CMS.CMSHelper.CMSContext.CurrentUser.UserName;
```

Checking if user is authenticated

[C#]

```
bool isAuthenticated = HttpContext.Current.User.Identity.IsAuthenticated;
```

Checking if user is member of a role

[C#]

```
if (CMS.CMSHelper.CMSContext.CurrentUser.IsInRole("CMSEditor", CMS.CMSHelper.
CMSContext.CurrentSiteName))
{
   //the current user is member of the CMSEditor role
}
```

Checking if a permission for a document is granted to a user

[C#]

```
if (CMS.CMSHelper.CMSContext.CurrentUser.IsAuthorizedPerTreeNode(nodeId, CMS.
TreeEngine.NodePermissionsEnum.Read) ==
      CMS.TreeEngine.AuthorizationResultEnum.Allowed)
{
   // the current user is authorized to read the document
}
```

Checking if a permission for a module (resource) is granted to a user

[C#]

```
if (CMS.CMSHelper.CMSContext.CurrentUser.IsAuthorizedPerResource("CMS.Forums",
"Modify"))
{
   // the "Modify" permission in the "CMS.Forums" module in the current website is
```

```
granted to the current user
}
```

Checking if a permission for a document type or custom table (class name) is granted to a user

[C#]

```
if (CMS.CMSHelper.CMSContext.CurrentUser.IsAuthorizedPerClassName("CMS.File",
"Delete"))
{
   // the "Delete" permission for all documents of type "CMS.File" in the current
website is granted to the current user
}
```

9.8 Site management, import and export

9.8.1 Creating a new website

The following example shows how you can create a new website based on the Blank website template:

[C#]

```
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.GlobalHelper;
using CMS.SiteProvider;
using CMS.CMSImportExport;

...

        // Site name
        string siteName = "testAPIsite";

        try
        {
            // Create site import settings
            SiteImportSettings settings = new SiteImportSettings();

            //Initialize the settings
            settings.SiteName = siteName;
            settings.SiteDisplayName = "Test API site";
            settings.SiteDescription = "Site for testing the API examples";
            settings.SiteDomain = "127.0.0.254";

            // Get 'Blank site' web template
            WebTemplateInfo template = WebTemplateInfoProvider.GetWebTemplateInfoByC

            // Template exists
            if (template != null)
            {
```

```
                // Set source file path
                string templatePath = template.WebTemplateFileName;
                settings.SourceFilePath = Server.MapPath(templatePath.TrimEnd('\\')


                // Load default selection to preselect the objects
                settings.LoadDefaultSelection();

                // Create new site using 'Blank' template
                ImportProvider.ImportObjectsData(settings);

                // Run site
                SiteInfoProvider.RunSite(siteName);

                this.lblInfo.Text = string.Format("New site with code name '{0}' has
                return;
            }
            else
            {
                this.lblInfo.Text = string.Format("Failed to create new site
'{0}'.<br />Web template 'Blank site' doesn't exist.", siteName);
                this.lblInfo.CssClass = "ErrorLabel";
            }
        }
        catch (RunningSiteException)
        {
            this.lblInfo.Text = string.Format("Failed to run site '{0}'.", siteName)
            this.lblInfo.CssClass = "ErrorLabel";
        }
        catch (Exception ex)
        {
            this.lblInfo.Text = string.Format("Failed to create new site '{0}'.<br
/>Original exception: " + ex.Message, siteName);
            this.lblInfo.CssClass = "ErrorLabel";
        }
```

9.8.2 Import and export of a website

You can export and import a website using Kentico CMS API, using the CMS.CMSImportExport. ImportProvider/ExportProvider classes. You can use the following methods:

- `public static void ImportSite(string siteName, string siteDisplayName, string siteDomain, string fullSourcePath, string websitePath)`
  - siteName - code name of the newly created website
  - siteDisplayName - display name of the newly created website
  - siteDomain - domain name of the newly created website
  - fullSourcePath - physical disk path of the package with exported site
  - websitePath - physical disk path of the website root

- `public static void ExportSite(string siteName, string fullExportFilePath, string websitePath, bool template)`
  - siteName - code name of the website to be exported
  - fullExportFilePath - physical disk path of the export package
  - websitePath - physical disk path of the website root
  - template - if false, a .zip file containing the site will be created under *fullExportFilePath* (should be

the full path including the .zip file name). Otherwise, the whole site will be exported to the same location (should be the full path without any file name in this case)

9.8.3 Getting website data

The following sample code shows how you can get a website as a *SiteInfo* object:

[C#]

```
using CMS.SiteProvider;

...

        string domainName = "localhost";
        string applicationPath = "/KenticoCMS";

        // Get site object by ID
        SiteInfo si = SiteInfoProvider.GetSiteInfo(1);

        // Get site object by site name
        SiteInfo si2 = SiteInfoProvider.GetSiteInfo("CorporateSiteASPX");

        // Get running site by domain name and application path
        SiteInfo si3 = SiteInfoProvider.GetRunningSiteInfo(domainName, applicationPa
```

The following sample code shows how you can get a DataSet containing all websites from the system:

[C#]

```
using System.Data;
using CMS.SiteProvider;

...

        string where = "";
        string orderby = "";

        // Get dataset of websites by where condition and order by
        DataSet ds2 = SiteInfoProvider.GetSites(where, orderby);

        // Get dataset of all websites
        DataSet ds = SiteInfoProvider.GetAllSites();
```

9.8.4 Updating website properties

The following example shows you how to modify website properties in your code:

[C#]

```
using CMS.CMSHelper;
using CMS.SiteProvider;
using CMS.WorkflowEngine;

...

        // Get current site name (you can use code name of the required site instead
        string siteName = CMSContext.CurrentSiteName;

        // Check if the site name is available
        if (siteName != null)
        {
            try
            {
                // Stop the site because we will change the domain of the site
                SiteInfoProvider.StopSite(siteName);

                // Get site info
                SiteInfo si = SiteInfoProvider.GetSiteInfo(siteName);

                if (si != null)
                {
                    si.DisplayName = "New display name";
                    si.Description = "New description";

                    // Change domain of the site
                    si.DomainName = "mynewdomain.com";

                    // Set new domain alias with default content language set to en-
                    SiteInfoProvider.AddDomainAlias(siteName, "newdomainalias", "en-

                    // Save the changes
                    SiteInfoProvider.SetSiteInfo(si);
                }

                // Run the site
                SiteInfoProvider.RunSite(siteName);

                lblInfo.Text = string.Format("Site '{0}' has been edited.", siteName
            }
            catch (RunningSiteException ex)
            {
                lblError.Text = "Site cannot be started.<br />Original exception: "
            }
            catch (Exception ex)
            {
                lblError.Text = "Error when modifying site properties.<br />Original
            }
        }
```

The following sample code shows how you can add a new content culture to an existing website:

[C#]

```
using CMS.SiteProvider;

...

        // Get site and culture objects
        SiteInfo si = SiteInfoProvider.GetSiteInfo("CorporateSiteASPX");
        CultureInfo ci = CultureInfoProvider.GetCultureInfo("AR-Sa");

        // If both exists
        if ((si != null) && (ci != null))
        {
            // Add culture to site
            CultureSiteInfoProvider.AddCultureToSite(ci.CultureID, si.SiteID);
        }
```

The following sample code shows how you can add a new domain alias to an existing website:

[C#]

```
using CMS.SiteProvider;

...

        // Get site object by site code name
        SiteInfo si = SiteInfoProvider.GetSiteInfo("CorporateSiteASPX");

        // If site exists
        if (si != null)
        {
            // Create new domain alias object
            SiteDomainAliasInfo sdai = new SiteDomainAliasInfo();

            // Set properties
            sdai.SiteDomainAliasName = "127.0.0.1";
            sdai.SiteID = si.SiteID;

            // Save to database
            SiteDomainAliasInfoProvider.SetSiteDomainAliasInfo(sdai);
        }
```

9.8.5 Starting and stopping websites

The following sample code shows how you can start a website:

[C#]

```
using CMS.SiteProvider;

...

        // Get site object by code name
        SiteInfo si = SiteInfoProvider.GetSiteInfo("CorporateSiteASPX");
```

```
        // If site exists
        if (si != null)
        {
            // Run site
            SiteInfoProvider.RunSite(si.SiteName);
        }
```

The following sample code shows how you can stop a website:

[C#]

```
using CMS.SiteProvider;

...

        // Get site object by code name
        SiteInfo si = SiteInfoProvider.GetSiteInfo("CorporateSiteASPX");

        // If site exists
        if (si != null)
        {
            // Stop site
            SiteInfoProvider.StopSite(si.SiteName);
        }
```

9.8.6 Deleting websites

The following example shows how you can delete an existing website; it uses a label control with ID *lblInfo* to display t

[C#]

```
using CMS.SiteProvider;
using CMS.CMSHelper;
using CMS.WorkflowEngine;

...

        // Code name of the site
        string siteName = "mysitecodename";

        // Delete records in CMS_Tree and CMS_Document
        DocumentHelper.DeleteSiteTree(siteName, null);

        // Delete other dependencies and the site
        SiteInfoProvider.DeleteSite(siteName);

        lblInfo.Text = string.Format("Site '{0}' has been deleted.", siteName);
```

9.9 Custom Providers

9.9.1 Custom Providers Overview

Kentico CMS allows you to develop the following custom providers and use them instead of the standard ones:

- Custom Data Provider
- Custom Search Provider
- Custom E-mail Provider
- Custom E-commerce Provider (the link leads to dedicated chapter in *Kentico CMS E-commerce guide*)

You can also use global event handlers to customize the system behavior, such as document modifications, authentication process, etc.

9.9.2 Custom Data Provider

The Custom Data Provider can be used to implement your own database connector.

Please note: The Custom Data Provider is NOT intended for accessing a non-Microsoft SQL Server database engines. It's only intended for minor modifications of the way the queries are executed against the Microsoft SQL Server.

1. Open the web project in Visual Studio.

2. Copy the **CustomDataProvider** project from *<installation directory>/CodeSamples* to your Solution directory.

3. Add the project **CustomDataProvider** to the solution.

4. Add the reference to the new project to the website project.

5. Add the reference to the **IDataConnectionLibrary**, **SettingsProvider**, **DirectoryUtilities** a **GlobalHelper** libraries located in the website project to the CustomDataProvider project.

6. Build the solution.

7. Add the following key to the AppSettings section of your **web.config** file:

```
<add key="CMSDataProviderAssembly" value="CMS.CustomDataProvider"/>
```

8. Run the website, it should use the **CustomDataProvider** library now.

9. If everything works fine, you can modify the code of the Custom Data Provider as needed.

9.9.3 Custom E-mail Provider

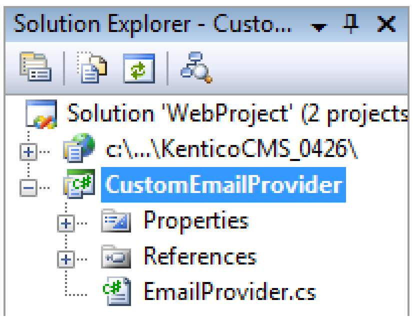
The custom e-mail provider allows you to use third-party e-mail components for sending e-mails or add custom actions when an e-mail is sent (e.g. logging the sent e-mails to some file for auditing purposes). All e-mails sent by Kentico CMS and its modules will use your custom e-mail provider.

Example

In this example, we will create a custom e-mail provider that will log every e-mail into a file.

1. Copy the **CustomEmailProvider** project from Kentico CMS installation (typically *C:\Program Files\KenticoCMS\<version>\CodeSamples\CustomEmailProvider*) to some development folder (not under the web project).

2. Open the **CMS** web project using the **WebProject.sln** file. Click **File -> Add -> Existing Project** and select the **CustomEmailProvider.csproj** file in the folder where you copied the **CustomEmailProvider** project. Your Solution Explorer window will look like this:

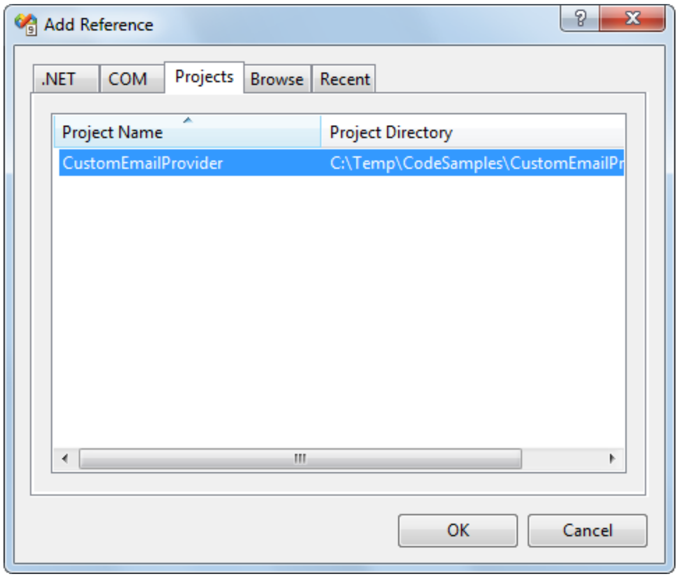


3. Unfold the **References** section of the **CustomEmailProvider** project and delete invalid references.

4. Right-click the **CustomEmailProvider** project and choose **Add Reference**. Choose the **Browse** tab and locate the **bin** folder of your CMS web project on the disk. Choose to add a reference to the following libraries:

- CMS.EmailProvider.dll
- CMS.IEmailEngine.dll
- CMS.SettingsProvider.dll

5. Unfold the **bin** folder in the **CMS web project**. Right-click the **bin** folder and choose **Add Reference**. Choose the **Projects** tab and click **OK** to add the **CustomEmailProvider** project reference:



6. Now you can modify the **CustomEmailProvider** library and place your code to the **SendEmail** method. Enter the following code into the **EmailProvider.SendEmail** method:

[C#]

```
// send the e-mail using the standard e-mail provider
CMS.EmailProvider.EmailProvider standardEmailProvider = new CMS.EmailProvider.
EmailProvider();
standardEmailProvider.SendEmail(siteName, message);

// log e-mail in the log file
System.IO.StreamWriter sw;
sw = System.IO.File.AppendText("C:\\_test\\EmailLogFile.txt"); // use a custom
valid path
sw.WriteLine(DateTime.Now.ToString() + ": " + message.Subject);
sw.Close();
```

7. Set the following value in your web.config file:

```
<add key="CMSEmailProviderAssembly" value="CMS.CustomEmailProvider" />
```

8. Click **Build -> Rebuild solution**. Go to the site and subscribe to a newsletter or use some other e-mail-related feature. When the e-mail is sent, it's logged in the file **EmailLogFile.txt** located in the path you specified in your code.

9.9.4 Custom Search Provider

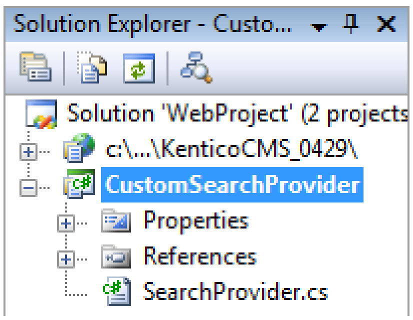
Kentico CMS allows you to write your own search provider that can use some third-party search engine. You can also write a custom search provider when you need to modify or filter search results returned by the standard search engine.

The following example explains how you can limit the standard search path to a subsection of your website.

Tip: This is only an example to show how the standard search engine can be modified. The search path can easily be modified by setting the **Path** property of individual search web parts or controls.

1. Copy the **CustomSearchProvider** project from Kentico CMS installation (typically *C:\Program Files\KenticoCMS\<version>\CodeSamples\CustomSearchProvider*) to some development folder (not under the web project).

2. Open the **CMS** web project using the **WebProject.sln** file. Click **File -> Add -> Existing Project** and select the **CustomSearchProvider.csproj** file in the folder where you copied the **CustomSearchProvider** project. Your Solution Explorer window will look like this:

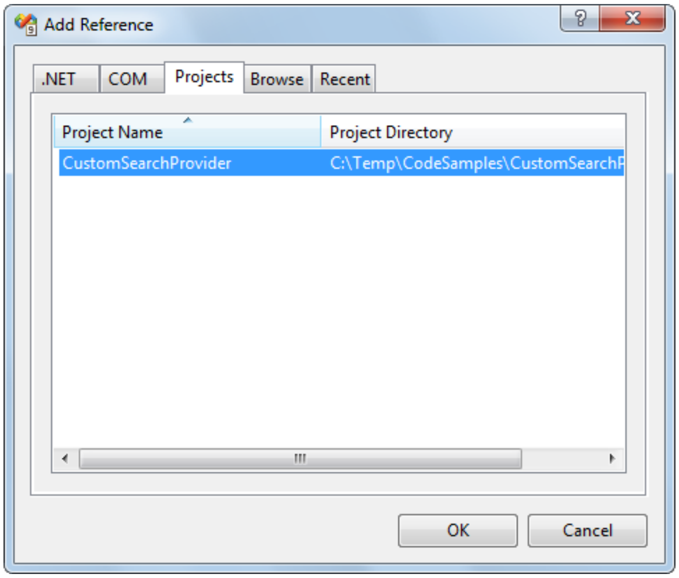


3. Unfold the **References** section of the **CustomSearchProvider** project and delete invalid references.

4. Right-click the **CustomSearchProvider** project and choose **Add Reference**. Choose the **Browse** tab and locate the **bin** folder of your CMS web project on the disk. Choose to add a reference to the following libraries:

- CMS.ISearchEngine.dll
- CMS.SearchProviderSQL.dll
- CMS.DataEngine.dll
- CMS.IDataConnectionLibrary.dll

5. Unfold the **bin** folder in the **CMS web project**. Right-click the **bin** folder and choose **Add Reference**. Choose the **Projects** tab and click **OK** to add the **CustomSearchProvider** project reference:



6. Now you can modify the **SearchProvider.cs** library and place your code into the Search method. Modify the code of the SearchProvider.Search method to look like this:

[C#]

```
public DataSet Search(string siteName, string searchNodePath, string cultureCode,
string searchExpression, SearchModeEnum searchMode, bool searchChildNodes, string
classNames, bool filterResultsByReadPermission, bool searchOnlyPublished, string
whereCondition, string orderBy, bool combineWithDefaultCulture)
{

    //limit searchNodePath to the /Products section of the website only
    searchNodePath = "/Products/%";

    CMS.SearchProviderSQL.SearchProvider standardSearchProvider = new CMS.
SearchProviderSQL.SearchProvider();

    DataSet ds = standardSearchProvider.Search(siteName, searchNodePath,
cultureCode, searchExpression, searchMode, searchChildNodes, classNames,
filterResultsByReadPermission, searchOnlyPublished, whereCondition, orderBy,
combineWithDefaultCulture);

    return ds;

}
```

7. Add the following value to the **/configuration/appSettings** section of your web.config file:

```
<add key="CMSSearchProviderAssembly" value="CMS.CustomSearchProvider" />
```

8. Click **Build -> Rebuild solution**. Go to the live site and try to search for some text using the SQL Search engine, for example on the **Corporate Sample site** at **Examples -> Web Parts -> Full-Text Search -> SQL Search dialog with results**. You will notice that the search now only returns results

from the **/Products** section of the website.

Please note

This example only modifies the SQL Search engine, the Smart Search engine remains unchanged.

9.10 Data layer

9.10.1 Overview

Kentico CMS has its own data layer components that ensure unified access to the database. Each entity, such as “user", “site", “document", has its own “data class". The data class represents the data structure of the entity (using the XML schema) and SQL queries for INSERT, UPDATE, SELECT and DELETE operations. The data layer is currently working only with Microsoft SQL Server, but it’s designed for use with other data engines in future.

How it Works

When you need to create a new user record, you create a new instance of DataClass type and specify its class name as “cms.user", which is a code name of this entity. The system automatically creates a new DataRow based on the user entity XML Schema. Then you set the values of its attributes, such as UserName or FullName and call the Insert method. The system uses pre-defined INSERT query to insert the values into the appropriate table that is also stored in the definition of the cms.user entity.

Related Namespaces

CMS.DataEngine
CMS.DataProviderSQL
CMS.IDataConnectionLibrary
CMS.SettingsProvider

Related Tables

CMS_Class
CMS_Query

9.10.2 Code examples

The following examples show how you can use the CMS.DataEngine library for low-level data manipulation.

Only for illustration

The code is used only for illustration. It's recommended that you use the **CMS.**

SiteProvider.UserInfoProvider class for manipulation of the user data.

Creating a new user

[C#]

```
using CMS.DataEngine;
...

DataClass userObj = new DataClass("cms.user");
userObj.SetValue("username", "johns");
userObj.SetValue("fullname", "John Smith");
userObj.Insert();
```

Selecting and updating an existing user

[C#]

```
using CMS.DataEngine;
...

DataClass userObj = new DataClass("cms.user", 10);
string userName = (string) userObj.GetValue("username");
userObj.SetValue("fullname", "John Smith Jr.");
userObj.Update();
```

Deleting a user

[C#]

```
using CMS.DataEngine;
...
DataClass userObj = new DataClass("cms.user", 10);
if (! userObj.IsEmpty())
{
   userObj.Delete();
}
```

Running a custom query

You can run a custom query if you first create it either manually in the CMS_Query table or through the administration interface (if it’s supported for the chosen entity).

[C#]

```
using CMS.DataEngine;
...
```

```
GeneralConnection cn = ConnectionHelper.GetConnection();
DataSet ds = null;
object[,] parameters = new object[1, 3];

parameters[0, 0] = "@UserName";
parameters[0, 1] = "johns";

ds = cn.ExecuteQuery("cms.user.selectbyname", parameters);
```

9.10.3 Pre- and post-processing queries

You can pre-process database queries and post-process query results using the **OnBeforeExecuteQuery** and **OnAfterExecuteQuery** events in the **SqlHelperClass**.

Pre-processing queries

The **OnBeforeExecuteQuery** event is executed before any query is executed. Using it, you can influence the behavior of the query and its code on the fly.

The code below is the delegate definition for the event:

```
/// <summary>
/// Query execution event handler
/// </summary>
/// <param name="query">Executed query</param>
/// <param name="conn">Connection</param>
public delegate void BeforeExecuteQueryEventHandler(QueryParameters query,
IDataConnection conn);
```

And this code example shows how you can use the event. You need to register the event in the **AfterApplicationStart** method in *~/App_Code/Global/CMS/CMSApplication.cs*. This particular example replaces 'CMS_User' with 'View_CMS_User' in case that the processed query is *cms.user.selectall*.

```
///
/// Fires after the application start event
///
public static void AfterApplicationStart(object sender, EventArgs e)
{
    // Add your custom actions
    CMS.SettingsProvider.SqlHelperClass.OnBeforeExecuteQuery += new CMS.
SettingsProvider.SqlHelperClass.BeforeExecuteQueryEventHandler
(BeforeExecuteQuery);
}

static void BeforeExecuteQuery(CMS.SettingsProvider.QueryParameters query, CMS.
IDataConnectionLibrary.IDataConnection conn)
{
    if (query.Name != null)
    {
        switch (query.Name.ToLower())
        {
```

```
            case "cms.user.selectall":
                query.Text = query.Text.Replace("CMS_User", "View_CMS_User");
                break;
        }
    }
}
```

Post-processing queries

The **OnAfterExecuteQuery** event is raised after any query is executed and you can use it to modify the result of the query. It can handle only a *DataSet*, thereby only calls using the *ExecuteQuery* method can be post-processed this way.

The code below is the delegate definition for this event:

```
/// <summary>
/// Query execution event handler
/// </summary>
/// <param name="query">Executed query</param>
/// <param name="conn">Connection</param>
public delegate void BeforeExecuteQueryEventHandler(QueryParameters query,
IDataConnection conn);
```

And this code example shows how you can use the event. You need to register the event in the **AfterApplicationStart** method in *~/App_Code/Global/CMS/CMSApplication.cs*. This particular example basically gives dynamically generated full name instead of the one that is set in the user settings. This will not be reflected in the UI as you are only processing the result of the query, so please take this just as an example of how you can use the event.

```
///
/// Fires after the application start event
///
public static void AfterApplicationStart(object sender, EventArgs e)
{
    // Add your custom actions
    CMS.SettingsProvider.SqlHelperClass.OnAfterExecuteQuery += new CMS.
SettingsProvider.SqlHelperClass.AfterExecuteQueryEventHandler(AfterExecuteQuery);
}

static void AfterExecuteQuery(CMS.SettingsProvider.QueryParameters query, CMS.
IDataConnectionLibrary.IDataConnection conn, ref DataSet result)
{
    if (query.Name != null)
    {
        switch (query.Name.ToLower())
        {
            case "cms.user.selectall":
                if (result != null)
                {
                    DataTable dt = result.Tables[0];
                    foreach (DataRow dr in dt.Rows)
                    {
```

```
                        dr["FullName"] = dr["FirstName"] + " " + dr["MiddleName"]
+ " " + dr["LastName"];
                    }
                }
                break;
        }
    }
}
```

9.11 Global events and their handling

9.11.1 Event handling overview

Global events allow you to execute custom actions when some CMS event occurs. For example, if a document is created in the CMS system, you can handle the corresponding event, get information about the new document and send its content by e-mail or use a third-party component to generate a PDF version of the document.

Events that can be handled

- Data updates - insert/update/delete actions for all data items.
- Exceptions
- Security events - authentication and authorization.
- Document events - TreeNode insert/update/delete operations.
- Workflow events - document approval/rejection/publishing/archiving operations.

Similar topics: Writing custom search engine provider

Enabling custom event handling

If you want to use custom event handlers, make sure the <*appSettings*> node of your web project's *web.config* file contains the following key:

```
<add key="CMSUseCustomHandlers" value="true" />
```

Default event handler library

The default custom event handler library is **CMS.CustomEventHandler**. Its name can be changed by means of a parameter in the <*appSettings*> node of your project's *web.config* file.

If the following line is inserted in the <*appSettings*> node, the library name will be changed to *CMS.CustomEventHandlerVB*:

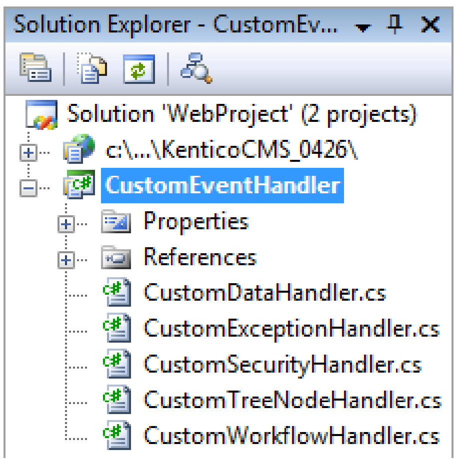
```
<add key="CMSCustomHandlersAssembly" value="CMS.
CustomEventHandlerVB" />
```

If you wanted to use this library in the following series of examples, it would require you to write the code in VB instead of C#.

How to write your own event handler

1. Copy the **CustomEventHandler** project from Kentico CMS installation (typically *C:\ProgramFiles\KenticoCMS\<version_number>\CodeSamples\CustomEventHandler*) to any custom folder that is not under the root of your web project (e.g. *C:\customhandler*).

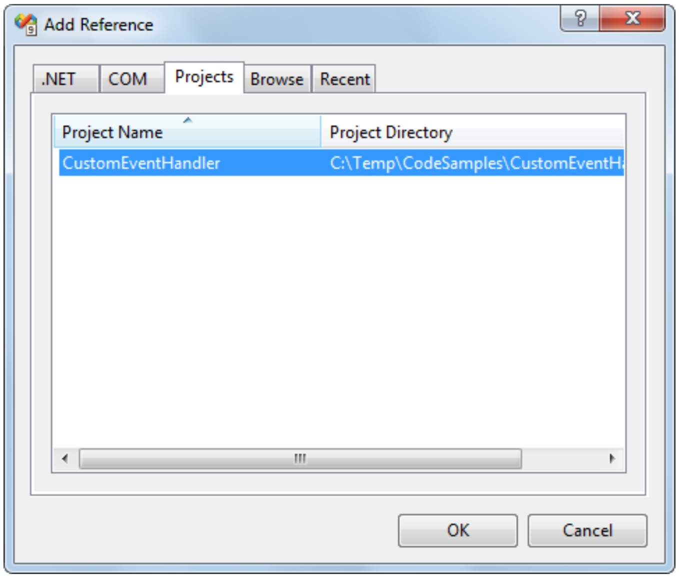
2. Open the **CMS** web project using the **WebProject.sln** file. Click **File -> Add -> Existing Project** and select the **CustomEventHandler.csproj** file located in the folder where you copied the **CustomEventHandler** project. Your Solution Explorer window should look like this:



3. Unfold the **References** section under the **CustomEventHandler** project and remove all invalid references (marked with the yellow exclamation icon).

4. Now you need to update the references. Unfold the **bin** folder in the Kentico CMS web project and delete the standard CMS.CustomEventHandler.dll library.

5. Right-click the **bin** folder and choose **Add Reference**. Choose the **Projects** tab and click **OK** to add a reference to the **CustomEventHandler** project:



6. Now right-click the **CustomEventHandler** project and choose **Add Reference**. Choose the **Browse** tab and locate the **bin** folder of your CMS web project on the disk. Choose to add a reference to the following libraries:

- CMS.GlobalEventHelper.dll
- CMS.TreeEngine.dll
- CMS.GlobalHelper.dll
- CMS.SiteProvider.dll
- CMS.CMSHelper.dll
- CMS.SettingsProvider.dll
- CMS.Staging.dll
- CMS.DataEngine.dll
- CMS.EmailEngine.dll
- CMS.WorkflowEngine.dll

You can also navigate to the **bin** folder on the **Browse** tab and copy&paste the following string into the **Add Reference** dialog to add all the libraries.

```
"CMS.GlobalEventHelper.dll" "CMS.TreeEngine.dll" "CMS.GlobalHelper.dll" "CMS.
SiteProvider.dll" "CMS.CMSHelper.dll" "CMS.SettingsProvider.dll" "CMS.Staging.dll"
"CMS.DataEngine.dll" "CMS.EmailEngine.dll" "CMS.WorkflowEngine.dll"
```

If you needed to use another library in your custom handlers, please add a reference to it the same way as described in this step.

7. Click **Build -> Rebuild solution**.

Now you can modify the libraries under the **CustomEventHandler** project and add your own code to handle the appropriate events. The places where you should place the code are marked with upper-case comments.

You can find more details on particular events in the following chapters.

9.11.2 Data handler (CustomDataHandler class)

This handler allows you to add custom actions to the following events:

- OnBeforeUpdate
- OnAfterUpdate
- OnBeforeInsert
- OnAfterInsert
- OnBeforeDelete
- OnAfterDelete
- OnGetContent

The events are applied to all data items that are stored to the database. It includes documents, user information or any other settings. They receive the data object of type **DataClass** that can be modified. In case of delete handlers, the class name and ID is provided.

Handling document events

For handling document events, please use the *CustomTreeNodeHandler* class instead of the *CustomDataHandler* class. Every document uses up to 3 tables, which leads to three separate events in the *CustomDataHandler* class.

Example

The following example shows how to handle the **OnAfterUpdate** event and send the password to the user whenever their profile is updated:

1. Open the **CustomDataHandler** class and put the following code at the beginning of the file. It adds the reference to the namespaces we will use to handle the event:

[C#]

```
using CMS.DataEngine;
using CMS.GlobalHelper;
using CMS.EmailEngine;
```

2. Put the following code inside the **OnAfterUpdate** method.

[C#]

```
// type the data object as DataClass
IDataClass dataItem = (IDataClass)dataObj;

// we want to handle only updates of user objects
if (dataItem.ClassName.ToLower() == "cms.user")
{
// we will use the CMS.EmailProvider to send e-mails
EmailMessage email = new EmailMessage();
email.From = "admin@domain.com";
```

```
// get the user's e-mail address
email.Recipients = ValidationHelper.GetString(dataItem.GetValue("Email"),
"admin@domain.com");
email.Subject = "Your password";
// get the user's password
email.Body = "Your password is:" + ValidationHelper.GetString(dataItem.GetValue(
"UserPassword"), "");
EmailSender.SendEmail(email);
}
```

Please note: getting user's password in the example above is only possible when passwords are stored in Plain text format; the setting is located in **Site Manager -> Settings -> Security**.

3. Set the **From** and **Recipients e-mail addresses** to your e-mail address.

4. If you don't have refference to *CMS.EmailEngine* in your *CustomEventHandler*, add it there.

5. Compile and run the project. Edit some user profile that uses your e-mail address. You should receive the e-mail message with your password.

9.11.3 Exception handler (CustomExceptionHandler class)

This class processes all **exceptions** that occur in the web application. You can use it to handle all exceptions and run custom actions, such as sending e-mail to administrator or logging the exception to you helpdesk or monitoring system.

The class has only one method: **OnException(Exception e)**

Example

The following example shows how to handle the OnException event and send the password to the administrator whenever an error occurs:

1. Open the **CustomExceptionHandler** class and put the following code at the beginning of the file. It adds the reference to the namespaces we will use to handle the event:

[C#]

```
using CMS.EmailEngine;
```

2. Put the following code inside the **OnException** method.

[C#]

```
// we will use the CMS.EmailProvider to send e-mails
EmailMessage email = new EmailMessage();
email.From = "admin@domain.com";
email.Recipients = "admin@domain.com";
email.Subject = "Exception";
```

```
email.Body = e.Message + "<br />" + e.StackTrace + "<br />" + e.Source;
EmailSender.SendEmail(email);
```

3. Set the From and Recipients e-mail addresses to you e-mail address.

4. Compile and run the project. Now, when some exception occurs or is raised by your code, the e-mail with exception details are sent.

9.11.4 Security handler (CustomSecurityHandler class)

The security handler allows you to integrate external user databases and modify the authentication and authorization process.

Any code added to the handlers is executed after the standard authentication or authorization checks performed by the system.

The class contains handlers for the following events:

- **OnAuthentication** - triggered when a user attempts to sign in with a name and password.
- **OnResourceAuthorization** - triggered when the system checks if a user is authorized to access a module.
- **OnUIElementAuthorization** - triggered when the system checks if a UI element should be displayed to a user.
- **OnClassNameAuthorization** - triggered when the system checks if a user is authorized to access a particular document type.
- **OnTreeNodeAuthorization** - triggered when the system checks if a user is authorized to access a document in the content tree.
- **OnFilterDataSetByPermissions** - triggered when a DataSet is filtered according to the permissions or custom personalization rules of the current user.

Example

In the following example, you will learn how to integrate external user authentication using the custom security handler.

The handler of the **OnAuthentication** event will be used for this purpose. It has the following parameters:

- **object userInfo** - an object representing the user attempting to log in. This object is returned as the result of the standard authentication check performed by the system. It is *null* if the default authentication failed.
- **string username** - a string containing the username entered during the login attempt.
- **string password** - a string containing the password entered during the login attempt.

The handler must return an object representing the user if external authentication using the entered credentials is successful, or *null* to indicate that authentication failed.

Now modify the code of the **OnAuthentication** handler according to the following:

[C#]

```
public override object OnAuthentication(object userInfo, string username, string
password)
{
    // Check if the user was authenticated by the system
    if (userInfo != null)
    {
        return userInfo;
    }

    // Sample external user credentials
    UserInfo usr = null;

    // Authenticate against the external source
    if ((username.ToLower() == "externaluser") && (password == "pass"))
    {
        // Create base user record if external authentication is successful
        usr = new UserInfo();
        usr.IsExternal = true;
        usr.UserName = username;
        usr.FullName = "externaluser fullname";
        usr.Enabled = true;

        // Initialize a hash table mapping roles to sites for the user
        Hashtable rolesTable = new Hashtable();

        // Get the code name of the current site
        string siteName = CMSContext.CurrentSite.SiteName;

        // Assign the user to the current site
        usr.SitesRoles[siteName.ToLower()] = rolesTable;

        // Add new role "external role" to the hash table to assign it to the user
        rolesTable["external role"] = 0;
    }

    // Return the user info object
    return usr;
}
```

For simplicity, the example does not use any particular database. Instead, it only checks if the current user name and password are equal to some constants. In a real-world scenario, you would need to replace this condition with code that checks if the user name with the given password is authenticated against your external database. Also, instead of simply assigning the user to a role named *external role*, you would have to implement code that checks the external database for any roles that the authenticated user is a member of and assigns them dynamically.

Once this is done, save the changes and **Build** the **CustomEventHandler** project. The system will now be able to perform authentication according to user data from an external source.

The roles created during this external authentication will not have any permissions assigned by default, so they will not authorize the user to perform any actions. You can programmatically check if a user belongs to a role using the **CMS.CMSHelper.CurrentUserInfo.IsInRole(string roleName, string siteName)** method and implement your own security logic in the other event handlers under the **CustomSecurityHandler** class.

However, we recommend importing all external roles into the **CMS_Role** table of the website's Kentico database. Then you can configure the appropriate permissions for these roles. This way, you will be able to fully use the built-in security model of Kentico CMS together with external users.

9.11.5 TreeNode handler (CustomTreeNodeHandler class)

The CustomTreeNodeHandler class allows you to execute custom actions when a document (TreeNode) is created, updated or deleted. It's useful if you need to synchronize changes to external systems, generate off-line version of the document in PDF, etc.

It handles the following events:

- OnBeforeInsert
- OnAfterInsert
- OnBeforeUpdate
- OnAfterUpdate
- OnBeforeDelete
- OnAfterDelete
- OnBeforeMove
- OnAfterMove
- OnBeforeCopy
- OnAfterCopy
- OnBeforeInsertNewCultureVersion
- OnAfterInsertNewCultureVersion

Example

The following example shows how to handle the OnAfterInsert event and send the newly added news item by e-mail:

1. Open the **CustomTreeNodeHandler** class and put the following code at the beginning of the file. It adds the reference to the namespaces we will use to handle the event:

[C#]

```
using CMS.TreeEngine;
using CMS.GlobalHelper;
using CMS.EmailEngine;
```

2. Put the following code inside the **OnAfterInsert** method.

[C#]

```
// type the document as TreeNode
TreeNode newsDoc = (TreeNode)treeNodeObj;

// handle the event only for news items
if (newsDoc.NodeClassName.ToLower() == "cms.news")
{
// get content of the inserted news item and send it by e-mail
EmailMessage email = new EmailMessage();
email.From = "admin@domain.com";
```

```
email.Recipients = "admin@domain.com";
email.Subject = ValidationHelper.GetString(newsDoc.GetValue("NewsTitle"), "");
email.Body = ValidationHelper.GetString(newsDoc.GetValue("NewsSummary"), "");
EmailSender.SendEmail(email);
}
```

3. Set the From and Recipients e-mail addresses to you e-mail address.

4. Compile and run the project. Create a new document of type News. You should receive the e-mail message with it text.

How to avoid neverending loops

If you need to call TreeNode.Update in the event handler (e.g. in the OnAfterUpdate event), you need to set **TreeProvider.UseCustomHandlers** property to false before calling the Update method.

9.11.6 Workflow handler

The workflow handler allows you to handle the following events:

- OnBeforeCheckOut - before document is checked out
- OnAfterCheckOut - after document is checked out
- OnBeforeCheckIn - before document is checked in
- OnAfterCheckIn - after document is checked in
- OnBeforeApprove - before document is approved
- OnAfterApprove - after document is approved
- OnBeforeReject - before document is rejected
- OnAfterReject - after document is rejected
- OnBeforePublish - before document is published
- OnAfterPublish - after document is published

When the document is published, the order of the events is following:

1. OnBeforeApprove
2. OnBeforePublish
3. OnAfterPublish
4. OnAfterApprove

Example

The following example shows how to send an e-mail with news document content when it's published:

1. Open the **CustomWorkflowHandler** class and put the following code at the beginning of the file. It adds the reference to the namespaces we will use to handle the event:

[C#]

```
using CMS.TreeEngine;
using CMS.GlobalHelper;
using CMS.EmailEngine;
using CMS.WorkflowEngine;
```

2. Put the following code inside the **OnAfterPublish** method:

[C#]

```
// type the document as TreeNode
TreeNode newsDoc = (TreeNode)treeNodeObj;

// handle the event only for news items
if (newsDoc.NodeClassName.ToLower() == "cms.news")
{
  // get content of the inserted news item and send it by e-mail
  EmailMessage email = new EmailMessage();
  email.From = "admin@domain.com";
  email.Recipients = "admin@domain.com";
  email.Subject = ValidationHelper.GetString(newsDoc.GetValue("NewsTitle"), "");
  email.Body = ValidationHelper.GetString(newsDoc.GetValue("NewsSummary"), "");
  EmailSender.SendEmail(email);
}
```

3. Set the From and Recipients e-mail addresses to you e-mail address.

4. Compile and run the project.

5. Configure your project so that it uses workflow for news items and create and publish a news item. You should receive the content of the news item by e-mail.

Getting the workflow step name

If you need to get the name of the workflow step (for example in the OnAfterApprove event), you need to use code like this:

[C#]

```
CMS.WorkflowEngine.WorkflowStepInfo previousStep = (CMS.WorkflowEngine.
WorkflowStepInfo) previousStepObj;
string stepCodeName = previousStep.StepName;
```

9.12 Customizing system objects with custom data or objects

In the example below, you will learn how to extend system objects with your own fields or data.

This extending is possible thanks to the **IRelatedData** interface and the **RelatedData** (object) property. If an object implements the **IRelatedData** interface, you can use its **RelatedData** (object) property to connect just about any object to it. The connected object is then shipped as a part of the system object.

On top of it, you can also make the related object behave as a native part of the system object and retrieve its properties as if they were properties of the system object. If the connected object implements the **IDataContainer** interface, it gets automatically connected to the system object's **GetValue** and **SetValue** methods.

Example

Let's presume that you want to customize the **SiteInfo** object with two additional properties and one method:

- **SiteOwner** - String, just the name of the site owner
- **SiteValidUntil** - DateTime until which the site is valid
- bool **IsValid**() - Returns true if the site is valid at the current time

1. First, you will need to create a custom class in the solution, containing the properties and the method as you can see in the code example below:

SiteRegistration.cs

[C#]

```
using System;
using CMS.SettingsProvider;

/// <summary>
/// Summary description for SiteRegistration
/// </summary>
public class SiteRegistration : IDataContainer
{
    #region "Variables"

    private string mSiteOwner = null;
    private DateTime mSiteValidUntil = DateTime.MinValue;

    #endregion


    #region "Properties"

    /// <summary>
    /// Gets or sets the site owner
    /// </summary>
    public string SiteOwner
    {
        get
        {
            return mSiteOwner;
        }
        set
        {
            mSiteOwner = value;
        }
    }
```

```
/// <summary>
/// Gets or sets the date until which the site is valid
/// </summary>
public DateTime SiteValidUntil
{
    get
    {
        return mSiteValidUntil;
    }
    set
    {
        mSiteValidUntil = value;
    }
}

#endregion


#region "IDataContainer members"

/// <summary>
/// Gets the column names
/// </summary>
public string[] ColumnNames
{
    get
    {
        return new string[] { "SiteOwner", "SiteValidUntil" };
    }
}


/// <summary>
/// Returns true
/// </summary>
/// <param name="columnName"></param>
public bool ContainsColumn(string columnName)
{
    switch (columnName.ToLower())
    {
        case "siteowner":
        case "sitevaliduntil":
            return true;

        default:
            return false;
    }
}


/// <summary>
/// Gets the object value
/// </summary>
/// <param name="columnName">Column name</param>
public object GetValue(string columnName)
```

```
    {
        switch (columnName.ToLower())
        {
            case "siteowner":
                return mSiteOwner;

            case "sitevaliduntil":
                return mSiteValidUntil;

            default:
                return null;
        }
    }


    /// <summary>
    /// Sets the field value
    /// </summary>
    /// <param name="columnName">Column name</param>
    /// <param name="value">New value</param>
    public bool SetValue(string columnName, object value)
    {
        switch (columnName.ToLower())
        {
            case "siteowner":
                mSiteOwner = (string)value;
                return true;

            case "sitevaliduntil":
                mSiteValidUntil = (DateTime)value;
                return true;

            default:
                return false;
        }
    }


    /// <summary>
    /// Tries to get the value from the object
    /// </summary>
    /// <param name="columnName">Column name</param>
    /// <param name="value">Returns the value</param>
    public bool TryGetValue(string columnName, out object value)
    {
        switch (columnName.ToLower())
        {
            case "siteowner":
                value = mSiteOwner;
                return true;

            case "sitevaliduntil":
                value = mSiteValidUntil;
                return true;

            default:
                value = null;
```

```
                return false;
            }
        }

        #endregion


        #region "Methods"

        /// <summary>
        /// Returns true if the site is valid
        /// </summary>
        public bool IsValid()
        {
            return this.SiteValidUntil > DateTime.Now;
        }

        #endregion
    }
```

2. Now that you have the class ready, you will want to bind it to the SiteInfo object. The binding will be dynamic, which means that the data will be loaded when it is requested. We will use the **OnLoadRelatedData** event which you can use for specific object types, including our SiteInfo objects. The event is part of the type information object referenced from the object info class.

What you need to do is to create the event and register it in the **AfterApplicationStart** event, both in the **~/App_Code/Global/CMS/CMSApplication.cs** class as you can see below:

[C#]

```
///
/// Fires after the application start event
///
public static void AfterApplicationStart(object sender, EventArgs e)
{
    // Add your custom actions
    CMS.SiteProvider.SiteInfo.TYPEINFO.OnLoadRelatedData += new CMS.
SettingsProvider.TypeInfo.ObjectLoadRelatedDataEventHandler
(SiteInfo_OnLoadRelatedData);
}

static object SiteInfo_OnLoadRelatedData(CMS.SettingsProvider.IInfoObject infoObj)
{
    // Get the related data from your external storage (make sure it implements
IDataContainer.      Here are just some data created on-the-fly.
    SiteRegistration sr = new SiteRegistration();
    sr.SiteOwner = "Martin Hejtmanek";
    sr.SiteValidUntil = DateTime.Now.AddDays(1);

    return sr;
}
```

3. Now try to paste the following code to the layout of some page on your site. Notice the bold parts of the code. As you can see, data stored in the custom properties are retrieved using the GetValue method

as if they were a native part of the SiteInfo object.

[C#]

```
<asp:Label runat="server" id="lblSiteInfo" />
<script runat="server">
  protected void Page_PreRender(object sender, EventArgs e)
  {
    CMS.SiteProvider.SiteInfo currentSite = CMSContext.CurrentSite;

    this.lblSiteInfo.Text = String.Format("Site '{0}' is valid until {1} and owned
by {2}.", currentSite.DisplayName, currentSite.GetValue("SiteValidUntil"),
currentSite.GetValue("SiteOwner"));
  }
</script>
```

As a result, you should see the following text on the page:

Site 'Corporate Site' is valid until 9/15/2009 1:09:50 PM and owned by Martin Hejtmanek.

4. It is also possible to get the values of the custom properties using macros. Try adding the **Text -> Static text** web part to your page and paste the following text to its **Text** property. The same text as in the previous example should be displayed on the live site.

```
Site '{%CMSContext.CurrentSite.SiteDisplayName%}' valid until {%CMSContext.
CurrentSite.SiteValidUntil%} and owned by {%CMSContext.CurrentSite.SiteOwner%}
```

9.13 Customizing the administration interface and web application events

This chapter describes how to customize the administration interface and website functionality in the way which is safe for future system updates and compatible with the import/export functionality.

Please note that a certain level of user interface customization can be achieved using UI personalization.

System events (Groups)

There are several groups of system events that can be handled by your custom code. There are predefined custom methods that allow you to influence default page cycle of the system pages/controls and customize the administration interface. These classes are located in *~/App_Code/Global/CMS* folder.

There are following system events that can be customized:

- CMSApplication.cs – Global application events (Start, End, …)
- CMSSession.cs – Session events (Start, End, …)
- CMSRequest.cs – Request events (Start, End, …)
- CMSPageEvents.cs – Page events (Init, Load, …)
- CMSUserControlEvents – User control events (Init, Load, …)

These events are fired on standard Application (Session, Request, Page, Control) cycle and provide you with the methods that are fired before and after each of these events.

- CMSCustom.cs – Common custom events. Special events that are raised from the system. Binding of these events is initialized within the Init method of this class.

Available methods to customize

CMSApplication.cs

- BeforeApplicationStart - Fires before the application start event
- AfterApplicationStart - Fires after the application start event
- BeforeApplicationEnd - Fires before the application end event
- AfterApplicationEnd - Fires after the application end event
- BeforeApplicationError - Fires before the application error event
- AfterApplicationError - Fires after the application error event

CMSSession.cs

- BeforeSessionStart - Fires before the session start event
- AfterSessionStart - Fires after the session start event
- BeforeSessionEnd - Fires before the session end event
- AfterSessionEnd - Fires after the session end event

CMSRequest.cs

- BeforeBeginRequest – Fires before the request start event
- AfterBeginRequest – Fires after the request start event
- BeforeEndRequest – Fires before the request end event
- AfterEndRequest – Fires after the request end event
- BeforeAcquireRequestState – Fires before the acquire request state event
- AfterAcquireRequestState – Fires after the acquire request state event
- BeforeAuthorizeRequest – Fires before the request authorization event
- AfterAuthorizeRequest – Fires after the request authorization event
- BeforeAuthenticateRequest – Fires before the request authentication event
- AfterAuthenticateRequest – Fires after the request authentication event

CMSPageEvents.cs

- BeforePagePreInit – Fires before page PreInit event
- AfterPagePreInit – Fires after page PreInit event
- BeforePageInit – Fires before page Init event
- AfterPageInit – Fires after page Init event
- BeforePageLoad – Fires before page Load event
- AfterPageLoad – Fires after page Load event
- BeforePagePreRender – Fires before page PreRender event
- AfterPagePreRender – Fires after page PreRender event
- BeforePageRender – Fires before page Render event
- AfterPageRender - Fires after page Render event

CMSUserControlEvents.cs

- BeforeUserControlInit – Fires before UserControl Init event

- AfterUserControlInit – Fires after UserControl Init event
- BeforeUserControlLoad – Fires before UserControl Load event
- AfterUserControlLoad – Fires after UserControl Load event
- BeforeUserControlPreRender – Fires before UserControl PreRender event
- AfterUserControlPreRender – Fires after UserControl PreRender event
- BeforeUserControlRender – Fires before UserControl Render event
- AfterUserControlRender - Fires after UserControl Render event

Examples

Following examples just briefly show how to customize the solution with your own functionality. If you need to customize certain page of the interface, you should see the code of the page so the customization matches the page structure.

Using the Page event to add tab (CMSPageEvents.cs) – This custom code will add a new tab to the main page of the E-commerce module:

[C#]

```
using CMS.UIControls;
using CMS.Controls;

/// <summary>
/// Fires before page Load
/// </summary>
/// <returns>Returns true if default action should be performed</returns>
public static bool BeforePageLoad(object sender, EventArgs e)
{
  // Add your custom actions
  CMSPage page = (CMSPage)sender;
  switch (page.RelativePath.ToLower())
   {
       case "/cmsdesk/tools/ecommerce/header.aspx":
           BasicTabControl tabControl = (BasicTabControl)page["TabControl"];
           // Add tabs
           string[,] tabs = BasicTabControl.GetTabsArray(1);

           tabs[0, 0] = "Google";
           tabs[0, 2] = "http://www.google.com";

           tabControl.AddTabs(tabs);

           break;
   }

   // Return true to allow the default Page_Load event
   return true;
}
```

Using the Custom event to resolve custom macro (CMSCustom.cs) – This custom code will handle the macro {#CurrentTime#} in all the modules that support custom macros.

[C#]

```
/// <summary>
/// Custom macro handler
/// </summary>
/// <param name="sender">Sender (active macro resolver)</param>
/// <param name="expression">Expression to resolve</param>
/// <param name="match">Returns true if the macro matches (was resolved)</param>
public static string ResolveCustomMacro(MacroResolver sender, string expression,
out bool match)
{
    match = false;
    string result = expression;

    // Add your custom macro evaluation
    switch (expression.ToLower())
    {
        case "currenttime":
            match = true;
            result = DateTime.Now.ToString();
        break;
    }

        return result;
}
```

9.14 Using API and CMS Controls outside CMS project

You can use Kentico CMS API and Kentico CMS Controls also outside the standard CMS website project. This chapter explains how you can configure your ASP.NET application so that it can use Kentico CMS API and Kentico CMS Controls.

Start Visual Studio and create a new ASP.NET application or open your existing ASP.NET web project.

Kentico CMS Initialization

Please note: you may need to initialize the application to be able to use all the features of Kentico CMS API. To do that, call the method **CMSContext.Init()** before any other calls to the API. You can do that at the time your project starts or anytime later, the method handles all necessary initializations of the environment.

Configuring the web.config file

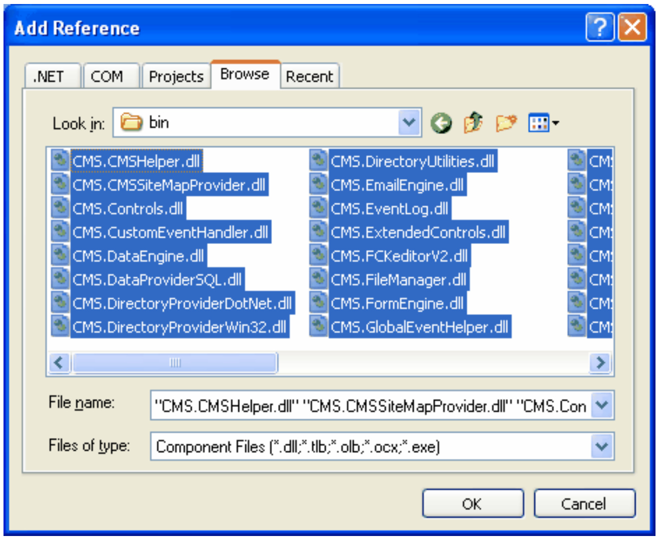
Add the connection string for the Kentico CMS database with name CMSConnectionString into the configuration/connectionStrings section of the web.config. The section will look like this:

```
<configuration>
  <connectionStrings>
    <add name="CMSConnectionString" connectionString="Persist Security
              Info=False;database=KenticoCMS;server=myserver;user id=sa;
password=mypassword;Current
              Language=English;Connection Timeout=120;" />
```

It's recommended that you copy the exact connection string line from the web.config file of the CMS web project.

Adding a reference to Kentico CMS API libraries

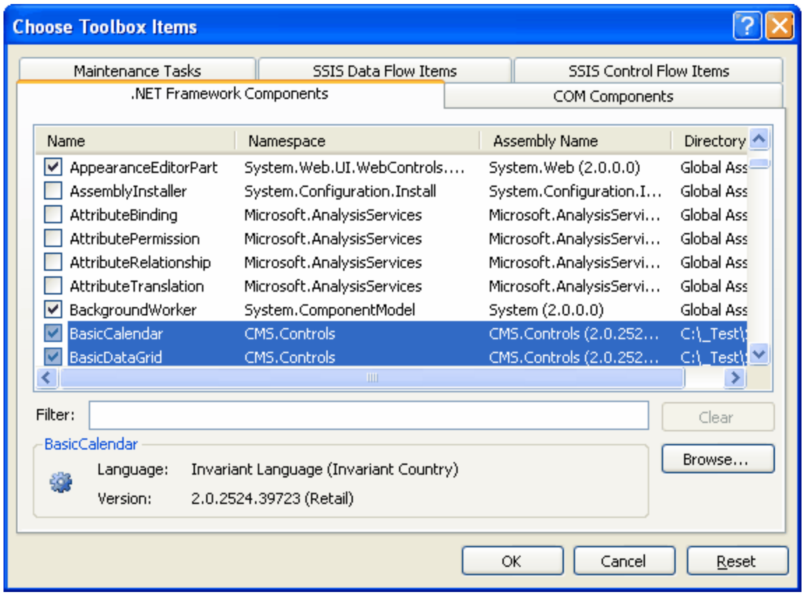
Now we will add a reference to Kentico CMS libraries. Right-click your web project in the **Solution Explorer** and and choose **Add reference...** Choose the **Browse** tab and locate the CMS web project's bin folder. Choose all libraries and click OK:



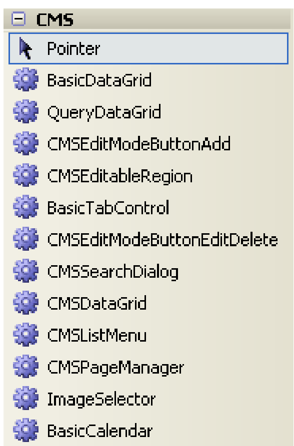
Adding Kentico CMS Controls to your toolbox

Now we will add Kentico CMS Controls to your toolbox so that you can easily drag-and-drop the controls on the web forms. Right-click the toolbox and choose **Add tab**. Use the name **CMS** and press Enter.

Right-click the new tab and click **Choose Items...** Click **Browse...** and locate the **CMS.Controls.dll** library in the **bin** folder of your application. Click **Open** and click **OK** on the **Choose Toolbox Items** dialog.



The controls are added to your tab:



Configuring your project for transformations (virtual path provider)

The transformations used by Kentico CMS Controls are retrieved using a virtual path provider. You need to add the following line to the `Application_Start` method in the *app_code\global.asax.cs/vb* file:

[C#]

```
CMS.VirtualPathHelper.VirtualPathHelper.RegisterVirtualPathProvider();
```

[VB.NET]

```
CMS.VirtualPathHelper.VirtualPathHelper.RegisterVirtualPathProvider()
```

If you cannot use the virtual path provider (e.g. in the medium trust environment), you may need to save the virtual objects to disk using the **Site Manager -> Administration -> System -> Deployment -> Save all virtual objects to disk** button and copy the folder **~/CMSTransformations** to the root of your own web project.

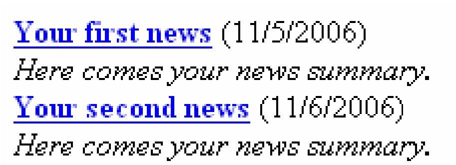
Using Kentico CMS Controls to display content from Kentico CMS database

Now that we have configured the web project for Kentico CMS, we will create a testing page that will display news items from Kentico CMS.

Create a new web form (ASPX page) in your custom project using Visual Studio. Drag and drop the CMSRepeater control on your page and set the following properties:

- ClassNames: cms.news
- CultureCode: en-us (or other language version)
- SiteName: CorporateSite (or other site code name)
- TransformationName: cms.news.preview
- SelectedItemTransformationName: cms.news.default

Run the project and navigate to the newly created page. You should see a page like this:



However, the links may not work at this moment since they are using the Kentico CMS web friendly URLs by default. So we need to modify the transformation cms.news.preview so that it points the user to same page, but with URL parameter **aliasPath** that will contain the page aliasPath. You can alternatively use your own URL parameters and set the Path property appropriately.

Go to **Kentico CMS Site Manager -> Development -> Document types -> edit News -> Transformations -> edit preview**. Change the following line of the transformation code:

```
<b><a href="<%# GetDocumentUrl() %>">
```

Like this:

```
<b><a href="?aliasPath=<%# Eval("NodeAliasPath") %>">
```

Save the changes. Since the transformations are cached, you need to restart (rebuild) your custom web application now so that the change is applied to your website.

Now you need to add a short code to the page code behind:

[C#]

```
protected void Page_PreInit(object sender, EventArgs e)
{
    if (Request.QueryString["aliaspath"] != null)
    {
        CMSRepeater1.Path = Request.QueryString["aliaspath"];
    }
    else
    {
        CMSRepeater1.Path = "/%";
    }
}
```

[VB.NET]

```
Protected Sub Page_PreInit(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.PreInit
    If Not Request.QueryString("aliaspath") Is Nothing Then
        CMSRepeater1.Path = Request.QueryString("aliaspath")
    Else
        CMSRepeater1.Path = "/%"
    End If
End Sub
```

Setting CMS control properties

Kentico CMS Controls load the content at early stage of the page life cycle. It means that if you want to set the properties programmatically, you need to set them in the Page_PreInit event.

If you need to set them for some reason later in the page life cycle, you need to call the `CMSRepeater1.ReloadData(true)` method so that the data is reloaded and the changes are applied.

Using Kentico CMS API to retrieve content from Kentico CMS database

Kentico CMS API can be used to script any action in Kentico CMS, including content retrieval and modification. The following example explains how you can retrieve documents from Kentico CMS database as a DataSet and display them using standard ASP.NET repeater control (instead of using the CMSRepeater control).

Add a new web form (ASPX page) to your web project. Drag and drop the Repeater control on the web form. Switch to the **Source** mode of the page and add the Following item template code inside the <asp:Repeater> control element:

[C#]

```
<ItemTemplate>
  <b><a href="?aliasPath=<%# Eval("NodeAliasPath") %>">
```

```
  <%# Eval("NewsTitle") %></a></b> (<%# ( (DateTime) Eval("NewsReleaseDate")).
ToString("d") %>)<br />
  <i><%# Eval("NewsSummary") %></i>
  <br />
</ItemTemplate>
```

[VB.NET]

```
<ItemTemplate>
  <b><a href="?aliasPath=<%# Eval("NodeAliasPath") %>">
  <%# Eval("NewsTitle") %></a></b> (<%# ( (DateTime) Eval("NewsReleaseDate")).
ToString("d") %>)<br />
  <i><%# Eval("NewsSummary") %></i>
  <br />
</ItemTemplate>
```

Switch to the code behind and add the following code to the beginning of the page:

[C#]

```
using CMS.TreeEngine;
using System.Data;
```

[VB.NET]

```
Imports CMS.TreeEngine
Imports System.Data
```

Add the following code inside the Page_Load method of the page:

[C#]

```
TreeProvider tp = new TreeProvider();
DataSet ds = tp.SelectNodes("CorporateSite", "/news/%", "en-us", true, "cms.news",
" NewsReleaseDate <= GetDate() ", " NewsReleaseDate DESC ", -1, true);
Repeater1.DataSource = ds;
Repeater1.DataBind();
```

[VB.NET]

```
Dim tp As New TreeProvider
Dim ds As DataSet = tp.SelectNodes("CorporateSite", "/news/%", "en-us", True,
"cms.news", " NewsReleaseDate <= GetDate() ", " NewsReleaseDate DESC ", -1, True)
Repeater1.DataSource = ds
Repeater1.DataBind()
```

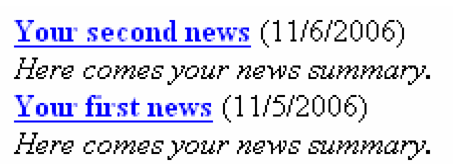
Please note

The code above works fine for just retrieving documents; if you want to perform some other operations with the documents (edit, delete, ...), you should initialize the tree provider with UserInfo, as shown below:

[C#]

```
UserInfo ui = UserInfoProvider.GetUserInfo("administrator");
TreeProvider tp = new TreeProvider(ui);
```

Run the website and navigate to your new page. You should see a page like this:



As you can see, you can use Kentico CMS API to retrieve content as a DataSet and display it with your custom .NET code.

9.15 Customizing providers from the App_Code folder

Custom providers (e.g. scheduled tasks) can be defined in *App_Code*, without the need to be compiled into an assembly. This can be achieved by defining a new class implementing the **ITask** interface in **App_Code/Global/CMS/CMSCustom.cs** and handle it in the **GetCustomClass** method.

You can find an example of a very simple *MyTask* class in *CMSCustom.cs*. The code of the task is displayed below.

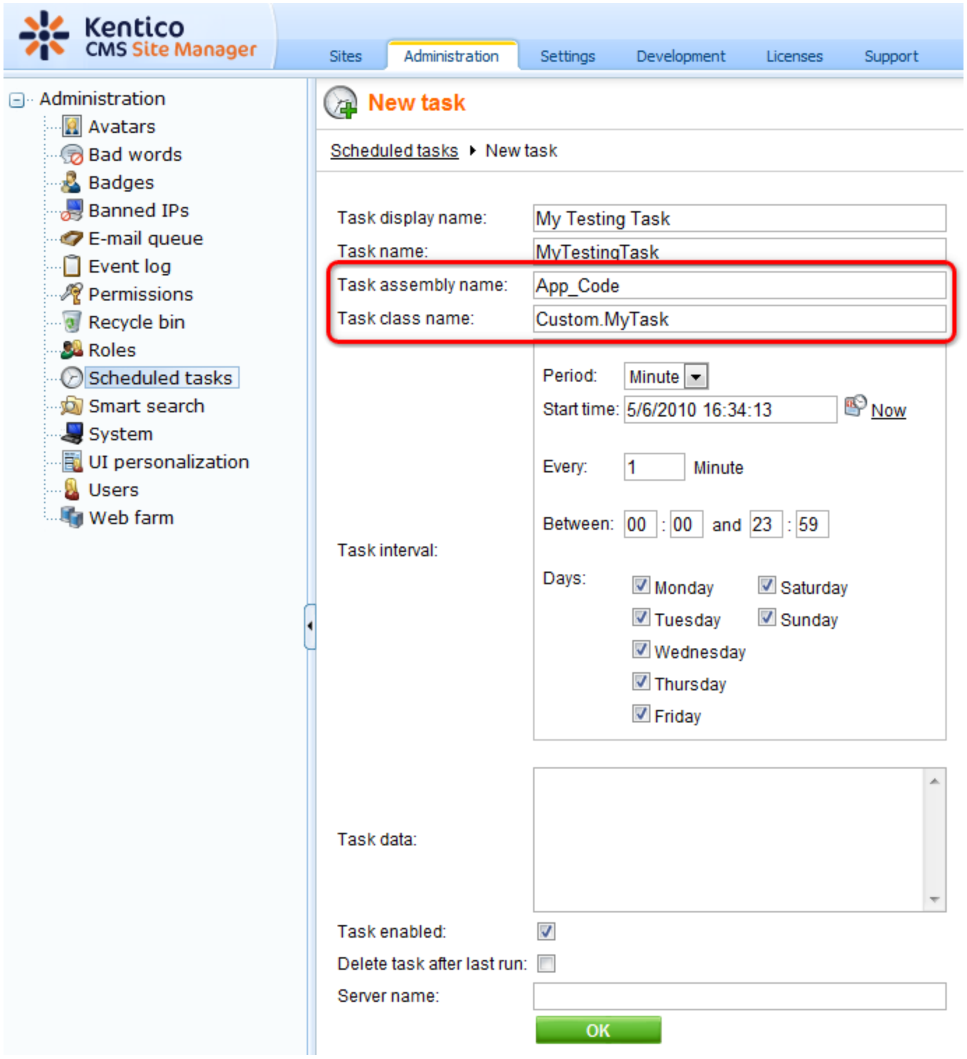
[C#]

```
/// <summary>
/// Gets the custom class object based on the given class name. This handler is
called when the assembly name is App_Code
/// </summary>
/// <param name="className">Class name</param>
public static object GetCustomClass(string className)
{
    // Provide your custom classes
    switch (className)
    {
        // Define the class MyTask implementing ITask and you can provide your
scheduled tasks out of App_Code
        case "Custom.MyTask":
            return new MyTask();
```

```
    }

    return null;
}

/// <summary>
/// Sample task class
/// </summary>
public class MyTask : ITask
{
    /// <summary>
    /// Executes the task
    /// </summary>
    /// <param name="ti">Task info</param>
    public string Execute(TaskInfo ti)
    {
        EventLogProvider ev = new EventLogProvider();
        ev.LogEvent("I", DateTime.Now, "MyTask", "Execute", null, "This task was
executed from '~/App_Code/Global/CMS/CMSCustom.cs'.");

        return null;
    }
}
```

To register the task in Kentico CMS, you need to create a new task in **Site Manager -> Administration -> Scheduled tasks**.

To ensure the required functionality, *App_Code* must be entered into the **Task assembly name** field and the class name handled in the *GetCustomClass* method needs to be entered into the **Task class name** field. With this task created, the code in the *MyTask* class is executed repeatedly after the set interval.


Kentico
CMS Site Manager
Sites
Administration
Settings
Development
Licenses
Support
Administration
Avatars
Bad words
Badges
Banned IPs
E-mail queue
Event log
Permissions
Recycle bin
Roles
Scheduled tasks
Smart search
System
UI personalization
Users
Web farm
New task
Scheduled tasks
New task
Task display name:
My Testing Task
Task name:
MyTestingTask
Task assembly name:
App_Code
Task class name:
Custom.MyTask
Period:
Minute
Start time:
5/6/2010 16:34:13
Now
Every:
1
Minute
Between:
00
00
and
23
59
Task interval:
Days:
Monday
Saturday
Tuesday
Sunday
Wednesday
Thursday
Friday
Task data:
Task enabled:
Delete task after last run:
Server name:
OK

Part X

Appendix A - Macro expressions

10 Appendix A - Macro expressions

This appendix describes macros that can be used within the Kentico CMS system. Macros are strings that are automatically resolved into their value equivalents and they represent a powerful option that ca often eliminate writing custom .NET code.

Types of macros

The macro type is determined by the character at the beginning and end of the macro. Macros look like:

{<**type character**><expression><parameters><**type character**>}

Where the **type character** can be one of:

- $ - Localization macro
- % - Context (data) macro
- ? - QueryString macro
- @ - Cookie macro
- # - Custom macro
- ^ - Control macro
- & - Path macro

Macros of the same type can be nested using the "index parentheses" syntax, e.g. *{**(1)**%FullName|(default){**(2)**%Username%**(2)**}%**(1)**}*. Click particular macro types in the listing above to get redirected to their detailed explanation.

Coverage of macros

The following table shows where particular types of macros can be used:

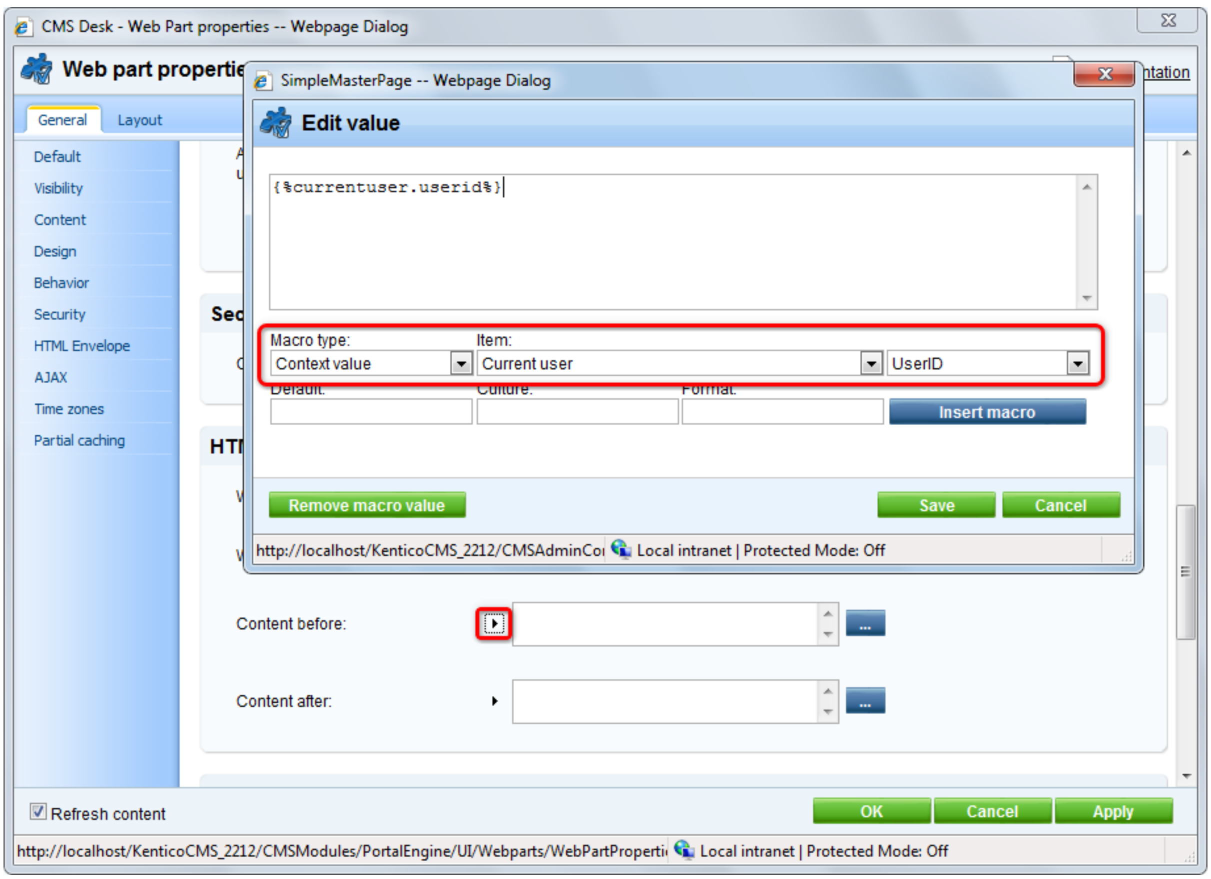
	Localization	Context	Query	Cookie	Custom	Path
BizForms	please see Modules -> BizForms -> Using macros with BizForms					
Document metadata	•	•	•	•	•	•
E-commerce	•	•	•	•	•	
Editable regions	•	•	•	•	•	
Forums	•					
Newsletters	•	•		•	•	•
Polls	•					
Reporting	•	•	•	•	•	•
Web part containers	•	•	•	•	•	•
Web part properties	•		•		•	

.NET code	Supported by API
Page layouts	Supported by API
Transformations	Supported by API

Entering macros

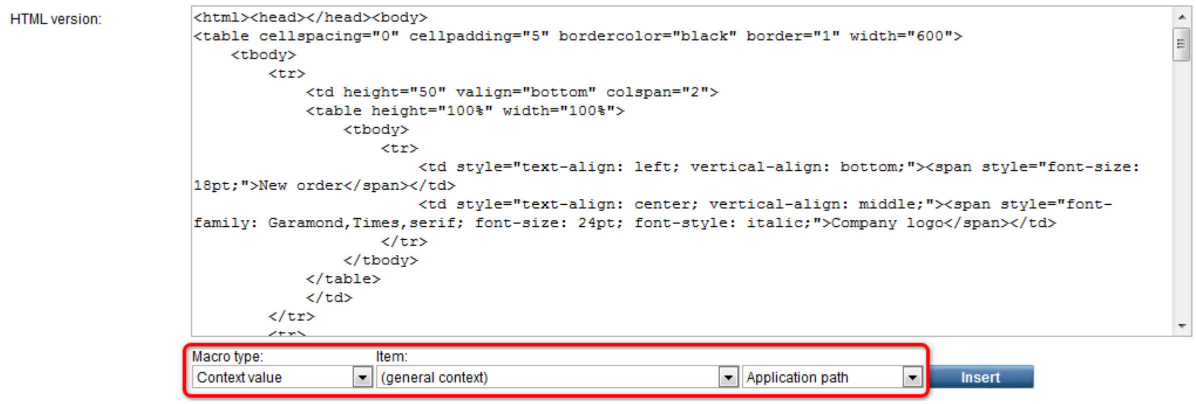
Macros can either be entered manually, or you can use the macro selector dialog, which allows entering of macros via drop-down lists and offers a complete range of available macros.

In web part properties, it can be accessed by clicking the little black arrow icon next to fields in web part properties, as you can see in the screenshot below. The expression can be selected by the highlighted drop-down lists, inserted into the field above using the **Insert** button and finally added as a value of the property using the **Save** button.



The same drop-downs are also available when creating or editing:

- **E-mail templates**
- **Newsletter issues**
- **Newsletter templates**
- **E-commerce invoice templates**



Localization macros

These macros are usually used on multilingual websites to help localize system strings. There are two types of locatization macros:

Basic

Basic localization macros are entered in the *{$string.key$}* format. The system uses the *ResHelper. GetString("string.key")* method and replaces the macro with the appropriate resource string.

<u>Example</u>: "*The weather is {$General.OK$}*" will be resolved as "*The weather is OK*".

In-place locatization

The string and its localized equivalents are included within the entered macro. The macro is entered in the *{$=Default string|cs-cz=Czech string|de-de=German string$}* format. It uses a localized string in case it is available or the default (first) string if it is not.

<u>Example</u>: "*The weather is {$=OK|cs-cz=dobre|de-de=gut$}*" will be resolved into "*The weather is dobre*" in Czech culture, "*The weather is gut*" in German culture and "*The weather is OK*" in any other culture.

Context (data) macros

This type of macros evaluates data based on current context. These macros can be used for example to parametrize web parts' parameters with current document or user values.

The format of the macro is *{%ColumnName%}* and the value is replaced with the appropriate value of the column based on current context. You can use all column names from **current site** data, **current user** data and **current document** data.

<u>Example</u>: "*Welcome {%FullName%}*" will be resolved as "*Welcome Global administrator*" when administrator is the current user.

There are also data macros with selectors to precisely identify the source of the data, these macros look like *{%CurrentDocument.DocumentName%}* and can be used for accurate determination of the data source. Availability of the selectors depends on the context where the macros are evaluated.

QueryString macros

These macros evaluate the query parameters information. These macros can be used for example to dynamically parametrize the controls by the querystring parameters.

The macros are entered in the *{?<querystring key>?}* format and it is replaced by the query parameter value.

<u>Example</u>: “*Current node ID: {?nodeid?}*” will be resolved as “*Current node ID: 10*” for a URL like “*default.aspx?nodeid=10*”

Cookie macros

These macros evaluate the cookie values. These macros can be used for example to parametrize the web parts with client-based persistent values like styles or user options.

The macro is entered in the *{@<cookie name>@}* format and it is replaced by the cookie value.

<u>Example</u>: *“Current style: {@StyleCookie@}*” will be resolved as “*Current style: Red*” if *StyleCookie* value is set to “*Red*”

Custom macros

These macros can be used to define your own macro. The macro is in the *{#Expression#}* format. When the macro is needed to be resolved, it calls the **ResolveCustomMacro** method located in the *~/App_Code/Global/CMS/CMSCustom.cs* class (or *~/Old_App_Code/Global/CMS/CMSCustom.cs* if you installed the project as a web application).

<u>Example</u>: “*Current time: {#CurrentTime#}*” will be resolved to “*Current time: 1/1/2008 10:30*“ with following custom macro handler:

[C#]

```
/// <summary>
/// Custom macro handler
/// </summary>
/// <param name="sender">Sender (active macro resolver)</param>
/// <param name="expression">Expression to resolve</param>
/// <param name="match">Returns true if the macro matches (was resolved)</param>
public static string ResolveCustomMacro(MacroResolver sender, string expression,
out bool match)
{
       match = false;
       string result = expression;

       // Add your custom macro evaluation
       switch (expression.ToLower())
       {
              case "currenttime":
              match = true;
              result = DateTime.Now.ToString();
              break;
```

```
        }

        return result;
}
```

Control macros

These macros can be used to resolve parts of text to the inline controls. The macro is in format *{^BizFormControl^}* and can (usually must) contain parameters for the control in a standard way of parametrized macros, such as *{^BizFormControl***|***(FormName)ContactForm^}*. It will be resolved to inline control which will get those parameters to initialize itself.

Path macros

These macros are entered in the *{&path&}* format. They can be used to resolve current document *Alias path* the same way as the *Path* property of controls. The macro is replaced by the resolved path.

Example: WhereCondition: *NodeAliasPath LIKE {&../%&}.* In this case, the macro will be resolved as parent document path and will result in selecting all siblings of the current document and their child documents. This macro is intended mostly for including the document structure context into the controls WHERE condition, but can be used for many more purposes.

See Appendix B - Path expressions for more details on how paths can be entered.

Macro parameters

It is possible to create macros with parameters to get better or specific functionality, especially for data (context) macros. Each parameter of the macro is separated with the "|" character located after the macro expression. You can use multiple macro parameters.

Examples: *{%SKUPrice***|(culture)en-us***%}*, *{%SKUPrice***|(culture)en-us|(format){0:f1}***%}*

The "|" character can be escaped using the "\|" sequence, e.g. *{%SKUPrice***|***(default)N***\|***A%}* will display "N|A", i.e. the "|A" sequence will not be interpreted as a new parameter.

Currently available parameters are:

- **Culture - "|(culture)<code>"** - saying that the specified culture should be used for the e.g. *{%SKUPrice|(culture)en-us%}* writes the product price in an English culture formatting
- **Format - "|(format)<format>"** – saying how the value should be formatted e.g. *{%SKUPrice|(format){0:f1}%}* writes the product price with precision for one decimal place
- **Default value - "|(default)<value>"** or **"|<value>"** - saying what should be returned when the value is not found. e.g. *{%SKUPrice|N/A%}* writes product price or N/A if the document is not a product
- **Encode – "|(encode)<true/false>"** – sprocesses the result with HTMLHelper.HTMLEncode e.g. *{%DocumentName|(encode)true%}* or *{%DocumentName|(encode)%}* writes the HTML encoded document name, such as *Black & white*; the default encoding settings can be enabled, in that

case, the settings may be |(encode)false to disable it

- **URLEncode – "|(urlencode)<true/false>"** – processes the result with HttpUtility.UrlEncode
  e.g. *{%DocumentName|(urlencode)true%}* or *{%DocumentName|(urlencode)%}* writes the URL encoded document name, such as *All%20items*; the default encoding settings can be enabled, in that case, the settings may be *|(urlencode)false* to disable it

- **ToLower – "|(tolower)<true>"** – converts the result to lowercase
  e.g. *{%DocumentName|(tolower)true%}* or *{%DocumentName|(tolower)%}* writes *black & white*

- **ToUpper - "|(toupper)<true>"** – converts the result to uppercase
  e.g. *{%DocumentName|(toupper)true%}* or *{%DocumentName|(toupper)%}* writes *BLACK & WHITE*

- **ToInt - "(toint)<default value>"** – converts the result to integer, if not successful, uses the default value
  e.g. *{?tagid|(toint)0?}* writes the valid *tagid* from querystring or *0* if not valid

- **ToBool - "(tobool)<default value>"** – conversion to Boolean, uses the (truevalue), (falsevalue) settings as a result, see below
  e.g. *{?onlyvalid|(tobool)true?}* writes false if *onlyvalid* querystring parameter is false, else returns true

- **ToGuid - "(toguid)<default value>"** – conversion to GUID
  e.g. **{?userguid|(toguid)?}** converts the userguid query parameter to *Guid* or *Guid.Empty*

- **ToDouble - "(todouble)<default value>"** – conversion to Double, uses the (culture) settings
  e.g. *{?price|(todouble)10.2?}* converts the price query parameter to double or 10.2

- **ToDateTime - "(todatetime)<default value>"** – conversion to DateTime, uses the (culture) settings
  e.g. *{?targettime|(todatetime)?}* converts the targettime query parameter to date time or *DateTime.MinValue*

- **ResolveBBCode - "(resolvebbcode)<true/false>"** – resolves the BB code in the result of the macro
  e.g. *{%MessageText|(resolvebbcode)true%}* or *{%MessageText|(resolvebbcode)%}* writes the resolved BB code such as conversion of [url]… to <a href="…

- **Equals - "(equals)<value>"** – returns "true" if the resolved value matches the given value, else returns "false"; uses the (truevalue), (falsevalue) settings for the output
  e.g. *{%UserName|(equals)administrator%}* writes true if the user is administrator

- **NotEquals - "(notequals)<value>" -** returns "false" if the resolved value matches the given value, else returns "true"; uses the (truevalue), (falsevalue) settings for the output
  e.g. *{%UserName|(notequals)administrator%}* – writes false if the user is administrator

- **TrueValue - "(truevalue)<value>"** – Output settings for the positive output of the comparison
  e.g. *{%UserName|(equals)administrator|(truevalue)Yes|(falsevalue)No%}* writes *Yes* if the user is administrator

- **FalseValue - "(falsevalue)<value>"** – Output settings for the negative output of the comparison
  e.g. *{%UserName|(equals)administrator|(truevalue)Yes|(falsevalue)No%}* writes *No* if the user is not administrator

API methods for evaluating macros

There is an easy way to resolve macros in .NET code. To resolve all the macros (recommended), use the following static method:

```
string CMS.CMSHelper.CMSContext.CurrentResolver.ResolveMacros(string inputText)
```

To resolve just the localization macros, use another static method:

```
string CMS.GlobalHelper.ResHelper.LocalizeString(string inputText)
```

Part XI

Appendix B - Path expressions

11 Appendix B - Path expressions

Many web parts and controls use a **Path** property that allows you to specify which documents should be displayed. This is the **AliasPath** property of the document. You can use either an exact path or you can use special characters for specifying multiple selection or relative paths:

Leaving the Path value empty

In many cases, **you can leave the Path value empty**. In this case, the Path value is set to the alias path of the currently displayed document.

In case of list controls/web parts, such as CMSRepeater/Repeater or CMSDataGrid/Grid, the path is set to *<current alias path>/%* if the current document is not of the same type as the required document in the **ClassNames** (document types) property. Otherwise, the path is set to the current alias path, which leads to automatic selection of the current document.

Using wildcard characters % and _

You can use **% as a wildcard character for any number of characters**, which allows you to select all documents under specified site section.

Examples:

/ - only root
/% - all documents

/Products - only the Products document.
/Products/% - all documents under the Products document.

You can also use **_ as a wildcard character for a single character**.

Examples:

/Product_ - selects documents /ProductA, /Product1, etc.

Using formatting string to get parts of the path

You can also use special expressions that **extract parts of the current path**, such as this.

Examples:

/{0}/{1}/% - all documents under the second level of the current path
/{0}/{1}/details - document Details under the second level of the current path

Using relative paths

You can use relative paths expressions to specify **sub-documents or parent documents**:

Examples:

./Product - document Product under the current path

../Product - document Product under the parent document of the current path
. - current path
.. - parent document of the current path
./% - all documents under the current path
../% - all documents under the parent document of the current path

Part XII

Appendix C - Web.config parameters

12 Appendix C - Web.config parameters

The system settings include appSettings keys and other settings, such as a connection string placed in appropriate sections of the web.config file. AppSettings keys are stored in the **/configuration/ appSettings** section.

The following setting categories are available:

- General settings
- Assembly settings
- Forbidden character settings in user and role names
- Content staging settings
- WYSIWYG editor settings
- E-commerce settings
- Event log settings
- File export settings
- Item listing settings
- Query string parameter name settings
- Transaction isolation settings
- Scheduler settings
- Security settings
- UI culture settings
- Web farm synchronization settings

General settings

Key	Description	Sample Value

CMSProgrammingLanguage	Indicates the programming language used in transformations and in custom code added to web parts.
	
	The default value is *C#*.
		`<add key="CMSProgrammingLanguage" value="C#" />`
		
		or
		
		`<add key="CMSProgrammingLanguage" value="VB" />`

CMSTrialKey	Contains a temporary trial license key. You can remove this value after installation. The default value is "" (empty string).	`<add key="CMSTrialKey" value="true" />`
CMSUseCustomHandlers	Indicates if custom handlers should be executed to process system events. See the Global events and their handling chapter for more details. The default value is *false*.	`<add key="CMSUseCustomHandlers" value="true" />`
CMSUseVirtualPathProvider	Indicates if the virtual path provider should be used (true by default). Before you	`<add key="CMSUseVirtualPathProvide`

	switch off the virtual path provider, please read the Pre-compilation (Publish function) chapter. The default value is *true*.	`r" value="false" />`
CMSShowWebPartCodeTab	Indicates if the Code tab is displayed in web part properties dialog in CMS Desk. This parameter can be used for backward compatibility purposes. Otherwise, using the **Code** tab is now obsolete. The default value is *false*.	`<add key="CMSShowWebPartCodeTab" value="true" />`
CMSShowWebPartBindingTab	Indicates if the Binding tab is displayed in web part properties dialog in CMS Desk. This parameter can be used for backward compatibility purposes. Otherwise, using the Binding tab is now obsolete. The dafault value is *false*.	`<add key="CMSShowWebPartBindingTab" value="true"/>`
CMSFullClientCache	Indicates if full client cache is enabled. If enabled, CSS styles and images will be cached in the client's browser. It is not recommended to use this option in non-live-site mode, as changes made to the pages would not be displayed due to the caching. The default value is *false*.	`<add key="CMSFullClientCache" value="true"/>`
CMSRenderGeneratorName	Indicates if the 'generator' meta tag stating that the page was generated by Kentico CMS is generated in the header of each page. The default value is *false*.	`<add key="CMSRenderGeneratorName" value="true"/>`
CMSClearFieldEditor	Determines field editor behavior when creating new fields. If true, new fields will have empty values of attributes. If false, new fields will have pre-defined values, the same as the previously selected field. The default value is *true*.	`<add key="CMSClearFieldEditor" value="true"/>`
CMSShowTemplateASPXTab	Indicates if ASPX code tab is displayed when editing a page template. Using this tab, ASPX code of a page template created using the Portal engine can be exported. The default value is *false*.	`<add key="CMSShowTemplateASPXTab" value="true"/>`
CMSDatabaseCulture	Specifies the default culture of the system's database.	`<add key="CMSDatabaseCulture"`

	The default value is *en-us*.	value="cz-cs"/>
CMSDisposeConnectionAfterClose	If true, database connection is automatically disposed (allocated resources released) when a database connection is closed. The default value is *false*.	<add key=" CMSDisposeConnectionAfterClose" value="true"/>
CMSUseSessionCookies	Indicates if session cookies are used or not. The default value is *true*.	<add key=" CMSUseSessionCookies" value="false"/>
CMSImportWindowsRoles	When Windows authentication is used and this key set to true, roles available in the Active Directory will be imported into the system. The default value is *true*.	<add key=" CMSImportWindowsRoles" value="true"/>
CMSFileScriptTimeout	The maximum number of seconds a script can run before the server terminates it. The default value is *7200*.	<add key=" CMSFileScriptTimeout" value="7200"/>
CMSUseExtensionOnPostback	When using friendly URL extensions, postback doesn't work in some cases. If you enable this setting, *.aspx* extension is attached to the URL in the form tag, which prevents the postback problems. The default value is *true*.	<add key=" CMSUseExtensionOnPostback" value="false"/>
CMSUseSQLResourceManager	If true, SQL Resource manager is used to retrieve strings used in the user interface. The default value is *true*.	<add key=" CMSUseSQLResourceManager" value="true"/>
CMSAllowCheckIOPermissions	If true, write permissions on the site folder are checked when necessary and produce an error message when they are insufficient. The default value is *true*.	<add key=" CMSAllowCheckIOPermissions" value="true"/>
CMSWorkflowSendEmailToModerator	If true, workflow notification e-mails will be sent to the user who is performing the current workflow step along with other users involved in the workflow. The default value is *false*.	<add key=" CMSWorkflowSendEmailToModerator" value="true"/>
CMSControlElement	If present in the web.config, the tag entered in the value will be used instead of the SPAN tag when generating pages.	<add key=" CMSControlElement" value ="div"/>

	The default value is *span*.	
CMSUseParsedSelfClose	Indicates if parsed self closing tags operations (faster) are used instead of standard self close filter. The default value is *true*.	`<add key="CMSUseParsedSelfClose" value="false"/>`
CMSGetFileEndRequest	If true, *ApplicationInstance. CompleteRequest()* is used instead of *Response.End()* in the *CompleteRequest* method. The default value is *true*.	`<add key="CMSGetFileEndRequest" value="true"/>`
CMSDefaultUserID	Specifies default user ID. The default value is *0*.	`<add key="CMSDefaultUserID" value="53"/>`
ImportFilesDiskPath	Specifies path to attachments that should be attached to documents imported via the SQL Import windows application. The default value is *""* (empty string).	`<add key="ImportFilesDiskPath" value="C:\Temp"/>`
CMSDeleteTemporaryAttachmentsOlderThan	Specifies how old should be attachments deleted by the 'Delete old temporary attachments' scheduled task. The value is entered in hours. Attachments older than the entered value will be deleted when the scheduled task is executed. The default value is *24*.	`<add key="CMSDeleteTemporaryAttachmentsOlderThan" value="12"/>`
CMSAllowGZip	Enables Gzip compression of output HTML code. The default value is *false*.	`<add key="CMSAllowGZip" value="true" />`
CMSDisableAdministrationInterface	Disables the administration interface. The 'Access denied' screen is displayed on each attempt to access CMS Desk or Site Manager. The default value is *false*.	`<add key="CMSDisableAdministrationInterface" value="true"/>`
CMSMaxNodeAliasLength	Maximal length of a document's node alias. The default value is 50 characters and larger names are trimmed. This key can be used to increase the length - useful for long node aliases. If you use the key, you also need to set the size of the 'NodeAlias' column in the 'CMS_Tree' database table to the same value. However, the maximal length of 'Node alias path' is 450 characters, so documents deep in the tree will have the	`<add key="CMSMaxNodeAliasLength" value="120" />`

	'Node alias path' trimmed anyway. The default value is 50.	
CMSMaxNodeNameLength	Maximal length of a document's node name. The default value is 100 characters and larger names are trimmed. This key can be used to increase the length - useful for long node names. If you use the key, you also need to set the size of the 'NodeName' column in the 'CMS_Tree' database table to the same value. The default value is *100*.	`<add key="CMSMaxNodeNameLength" value="150" />`
CMSWebAnalyticsSlidingIPExpiration	This key is used when the *Site Manager -> Settings -> Web Analytics -> Remember visitors by IP (minutes)* key has a value higher than 0. If *enabled* (by default), users who are active on the site but have disabled cookies are not logged as new visitors after the set time. If *disabled*, even an active user with disabled cookies is logged as a new site visitor after the set time.	`<add key="CMSWebAnalyticsSlidingIPExpiration" value="false" />`
CMSAuthenticationType	This key overrides the values returned by the *IsFormsAuthentication* and *IsWindowsAuhentication* methods in *CMS.GlobalHelper.RequestHelper*. You will typically use this if you are using a custom authentication provider whose authentication type is a non-standard one (e.g. Federated authentication) to make Kentico CMS handle it as if it was windows or forms authentication. The following values are available: • **default**: standard behavior • **forms**: *IsFormsAuthentication* always returns *true*, *IsWindowsAuthentication* always returns *false* • **windows**: *IsFormsAuthentication* always returns *false*, *IsWindowsAuthentication* always returns *true* • **both**: *IsFormsAuthentication* always returns *true*, *IsWindowsAuthentication* always returns *true*	`<add key="CMSAuthenticationType" value="windows" />`

	The default value is *default*.	

Special settings for assemblies

You can use the following keys to specify which assemblies are used:

Key	Description	Sample Value

CMSDirectoryProviderAssembly	Name of the assembly that should be used for operations in the file system. You can choose from the following options:
	• **CMS.DirectoryProviderDotNet** - this is a managed code library that uses System.IO methods for disk operations. It is useful for environment that allows only managed code.
	• **CMS.DirectoryProviderWin32** - this is an unmanaged code library that uses Win32 API for disk operations. It is useful for environment that requires Win32 API calls.
	
	The default assembly is *CMS. DirectoryProviderDotNet*.
		`<add key="CMSDirectoryProviderAssembly" value="CMS.DirectoryProviderDotNet" />`
		
		or
		
		`<add key="CMSDirectoryProviderAssembly" value="CMS.DirectoryProviderWin32" />`

CMSSearchProviderAssembly	Name of the assembly that is used for full-text search. The default assembly is *CMS. SearchProviderSQL*.	`<add key="CMSSearchProviderAssembly" value="CMS.CustomSearchProvider" />`
CMSDataProviderAssembly	Specifies custom data provider assembly used as the database connector for the CMS. See this topic for more details. The default assembly is *CMS. DataProviderSQL*.	`<add key="CMSDataProviderAssembly" value="CMS.CustomDataProvider"/>`
CMSEmailProviderAssembly	Specifies custom email provider assembly. Read this topic for more information. The default assembly is *CMS. EmailProvider*.	`<add key="CMSEmailProviderAssembly" value="CMS.CustomEmailProvider"/>`
CMSCustomHandlersAssembly	Specifies custom event handler assembly. Read this topic for more information. The default assembly is *CMS. CustomEventHandler*.	`<add key="CMSCustomHandlersAssembly" value="CMS.CustomEventHandler"/>`
CMSVirtualPathProviderAssem	Specifies the Custom virtual path provider	`<add key="`

bly	assembly. See this topic for more details. The default assembly is *CMS. VirtualPathProvider*.	CMSVirtualPathProviderAssembly" value="CMS.CustomVirtualPathProvider"/>
CMSCustomEcommerceProviderAssembly	Specifies the custom e-commerce provider assembly. See this topic for more details. The default assembly is *CMS. EcommerceProvider*.	<add key="CMSCustomEcommerceProviderAssembly" value="CMS.CustomECommerceProvider" />

Special settings for forbidden characters in user and role names

You can use the following keys to configure forbidden characters in user and role names:

Key	Description	Sample Value
CMSEnsureSafeUserNames	Indicates if forbidden characters in user names imported from AD should be replaced. If turned off by setting the value to *false*, you may experience problems when editing users imported with forbidden characters. Therefore, it is NOT RECOMMENDED to turn it off. The default value is *true*.	<add key="CMSEnsureSafeUserNames" value="false" />
CMSEnsureSafeRoleNames	Indicates if forbidden characters in role names imported from AD should be replaced. If turned off by setting the value to *false*, you may experience problems when editing roles imported with forbidden characters. Therefore, it is NOT RECOMMENDED to turn it off. The default value is *true*.	<add key="CMSEnsureSafeRoleNames" value="false" />
CMSForbiddenUserNameCharactersReplacement	Sets the character by which forbidden characters in user names imported from AD should be replaced. If not set, value from *Site Manager -> Settings -> URLs -> Forbidden characters replacement* is used.	<add key="CMSForbiddenUserNameCharactersReplacement" value="-" />
CMSForbiddenRoleNameCharactersReplacement	Sets the character by which forbidden characters in role names imported from AD should be replaced. If not set, value from *Site Manager -> Settings -> URLs -> Forbidden characters replacement* is used.	<add key="CMSForbiddenRoleNameCharactersReplacement" value="-" />

Special settings for Content staging

You can use the following keys to achieve specific behavior of the Content staging module:

Key	Description	Sample Value
CMSStagingAcceptAllCertificates	Causes all certificates will be accepted when performing content staging tasks through SSL (X.509). If false, only certificates generated by a certification authority will be accepted. The default value is *false*.	`<add key="CMSStagingAcceptAllCertificates" value="true" />`
CMSStagingUseTreeCustomHandlers	Ensures that events will be raised in *CustomTreeNodeHandler* when performing content staging synchronization. The *CMSUseCustomHandlers* key needs to be enabled too for this to work. The default value is *false*.	`<add key="CMSStagingUseTreeCustomHandlers" value="true" />`
CMSMediaFileMaxStagingSize	Sets the maximal file size of synchronized media files in kiloBytes. Files in media libraries that are larger than the entered value will not be synchronized using the Content staging module. The default value is *2147483647 (int. MaxValue)*.	`<add key="CMSMediaFileMaxStagingSize" value="1024" />`
CMSStagingTreatServerNamesAsInstances	Global metadata changes such as changes to document types, custom tables and system tables produce staging tasks for all staging servers of all sites. In such case, it is recommended to synchronize such changes to all servers of all sites at the same time to prevent overwriting of such metadata and loosing the data by synchronizing the older tasks later. You can use this key to make sure that once the global task is synchronized, it is deleted from all other servers with the same name to prevent such possibility. The default value is *false* since staging can use multiple target instances targeted with the same names.	`<add key="CMSStagingTreatServerNamesAsInstances" value="true" />`

Special settings for the WYSIWYG editor

You can use the following keys to configure the WYSIWYG editor:

Key	Description	Sample Value
CMSWYSIWYGFixXHTML	Indicates if the WYSIWYG editor should automatically try to fix XHTML incompatibilities in the code it generates. Supported values are *true* and *false*. The default value is *true*.	`<add key=" CMSWYSIWYGFixXHTML" value="true" />`
CMSLoadHTMLEditorByIFrame	If true, the WYSIWYG editor is loaded in an iFrame.	`<add key=" CMSLoadHTMLEditorByIFrame" value="true"/>`
FCKeditor:BasePath	Specifies the location of the WYSIWYG editor (FCK editor). By default, it is located in ~/ *CMSAdminControls/FCKeditor*.	`<add key="FCKeditor: BasePath" value="/ FCKeditor/"/>`
FCKEditor:UserFilesPath	Specifies path to user files. The default path is */UserFiles/*.	`<add key="FCKEditor: UserFilesPath" value="/ UserFiles/"/>`
FCKEditor: PersonalizeToolbarOnLiveSite	Indicates if the FCK toolbar can be personalized on the live site. See this topic for more details. Supported values are *true* and *false*. The default value is *false*.	`<add key="FCKEditor: PersonalizeToolbarOnLiveSite" value="true" />`

Special settings for E-commerce

You can use the following keys when you need to achieve specific behavior of the E-commerce module. The keys are also explained in E-commerce guide -> Configuration settings -> Web.config settings.

Key	Description	Sample Value
CMSShoppingCartExpirationPeriod	Number of days after which E-commerce shopping cart content is deleted from the database. It's used for deleting unused shopping carts of anonymous users that are stored in the database with ID stored in the browser cookie. The default value is *30*.	`<add key=" CMSShoppingCartExpirationPeriod" value="60" />`
CMSUseCurrentSKUData	Indicates if the E-commerce module should use the price from the existing order items or from the current SKU data when re-calculating the order. The default value is *false*.	`<add key=" CMSUseCurrentSKUData" value="true"/>`
CMSEnableOrderItemEditing	If true, order item parameters, such as order item name and order item unit price, can be modified when editing an existing order.	`<add key=" CMSEnableOrderItemEditing" value="true"/>`

	The default value is *false*.	
CMSUseMetaFileForProductImage	Indicates if meta files should be used for product images in E-commerce. The default value is *true*.	<add key="CMSUseMetaFileForProductImage" value="true"/>
CMSUseCustomEcommerceProviders	Specifies whether to use custom E-commerce provider handlers. See this topic for more details. The default value is *false*.	<add key="CMSUseCustomEcommerceProviders" value="true"/>

Special settings for event logging

You can use the following keys when you need to achieve specific behavior of the Event log:

Key	Description	Sample Value
CMSLogEvents	Indicates if logging events in the Event log is enabled. The default value is *true*.	<add key="CMSLogEvents" value="true"/>
CMSLogMetadata	Indicates if logging of metadata events in the Event log is enabled. The default value is *false*.	<add key="CMSLogMetadata" value="true"/>
CMSLogKeepPercent	Coefficient for Event log deletion, keeps the specified percentage of log items alive and deletes the log by batch when the set percentage is exceeded. The default value is *10*.	<add key="CMSLogKeepPercent" value="15"/>
CMSLogEventsToFile	If true, events are also logged into the ~\App_data\logevents.log file. The default value is *false*.	<add key="CMSLogEventsToFile" value="true"/>
CMSLogFieldChanges	If true, details about particular object changes are logged with the changes in Event log. The default value is *false*.	<add key="CMSLogFieldChanges" value="true"/>
CMSLogDocumentFieldChanges	If true, details about particular document changes are logged with the changes in Event log. The default value is *false*.	<add key="CMSLogDocumentFieldChanges" value="true"/>

Special settings for file exporting

You can use the following keys when you want to configure file exporting:

Key	Description	Sample Value
CMSExportExcludedFolders	Specifies which folders will be filtered from being included in Files folder of the export package. **.svn** folders are excluded by default, even without this key added. More info here.	`<add key="CMSExportExcludedFolders" value="test*;cms*" />`
CMSExportExcludedFiles	Specifies which files will be filtered from being included in the Files folder of the export package. **.scc** files are excluded by default, even without this key added. More info here.	`<add key="CMSExportExcludedFiles" value="test*;cms*" />`

Special settings for item listing

You can use the following keys when you want to configure item listing:

Key	Description	Sample Value
CMSDefaultListingFilterLimit	Required number of items listed in a listing across the whole UI for a filter to be displayed. If the number of listed items is lower than this value, the filter is not displayed. If it is larger, the filter is displayed. The default value is *25*.	`<add key="CMSDefaultListingFilterLimit" value="40" />`
CMSDefaultListingPageSize	Initial page size (the *Items per page* setting) of listings across the whole UI. The default value is *25*.	`<add key="CMSDefaultListingPageSize" value="50" />`
ShowFirstLastButtons	If enabled, the first and last page buttons are included in pagers of listings with large numbers of items in the UI. If disabled, the control is hidden in the listings. If both this and the *ShowDirectPageControl* keys are disabled, only *TopN* items are loaded, while *TopN* = *PageSize * (currentPageIndex + CurrentPagesGroupSize)*. The default value is *true*.	`<add key="ShowFirstLastButtons" value="false" />`
ShowDirectPageControl	If enabled, direct page control is included in pagers of listings with large	`<add key="ShowDirectPageControl"`

	numbers of items in the UI. If disabled, the control is hidden in the listings. If both this and the *ShowFirstLastButtons* keys are disabled, only *TopN* items are loaded, while *TopN = PageSize * (currentPageIndex + CurrentPagesGroupSize)*. The default value is *true*.	`value="false" />`

Special settings for query string parameter names

You can use the following keys to change certain query string parameter names:

Key	Description	Sample Value
CMSLanguageParameterName	Changes the name of the *lang* query string parameter so that you get *Home?sprache=de-de* instead of the default *Home?lang=de-de*. The default value is *lang*.	`<add key="CMSLanguageParameterName" value="sprache" />`
CMSAliasPathParameterName	Changes the name of the *aliaspath* query string parameter so that you get *products.aspx?ap=/products/myproduct* instead of the default *products.aspx?aliaspath=/products/myproduct*. The default value is *aliaspath*.	`<add key="CMSAliasPathParameterName" value="ap" />`

Special settings for transaction isolation

You will use the following settings only in special cases when you encounter problems with deadlocks when updating a document:

Key	Description	Sample Value
CMSTransactionIsolationLevel	Isolation level for explicit transactions. The default value is *ReadUncommitted*.	`<add key="CMSTransactionIsolationLevel" value="ReadCommitted" />`
CMSDefaultIsolationLevel	Isolation level for all queries, even without transactions. The default value is *ReadUncommitted*.	`<add key="CMSDefaultIsolationLevel" value="ReadUncommitted" />`
CMSUseDefaultIsolationLevelOnlyWithOpenTransactions	If true, the default isolation level is used only when some transaction is already running. The default value is *true*.	`<add key="CMSUseDefaultIsolationLevelOnlyWithOpenTransactions" value="true" />`

CMSAllowSimultaneousTransactions	If false, there can be only one transaction running at the same time. The default value is *true*.	`<add key=" CMSAllowSimultaneousTransactions" value="true" />`
CMSMaxTransactionWaitTimeout	Indicates how many seconds a transaction should wait for completion of another running transaction if simultaneous transactions are not allowed. The default value is *1*.	`<add key=" CMSMaxTransactionWaitTimeout" value="5" />`

Special settings for the scheduler

By adding the following keys to your web.config, you can configure the Scheduler:

Key	Description	Sample Value
CMSUseAutomaticScheduler	Indicates if automatic scheduling should be used. Automatic scheduler periodically requests the **cmspages/scheduler.aspx** page which ensures that scheduled tasks are processed automatically. If turned off (*false* - by default), the tasks are processed at the end of page requests. If it's *true*, the *cmspages/scheduler.aspx* page is requested periodically.	`<add key=" CMSUseAutomaticScheduler " value="false" />`
CMSRunSchedulerWithinRequest	If *true* (the default value), the scheduler is executed within the standard EndRequest of a page. If *false*, the scheduler is executed via the cmspages/scheduler.aspx page.	`<add key=" CMSRunSchedulerWithinRequest " value="false" />`
CMSSchedulerURL	URL of the physical location of the scheduler.aspx page. The default value is *~/cmspages/scheduler.aspx*	`<add key=" CMSSchedulerURL" value=" https://domain/cmspages/scheduler.aspx" />`
CMSSchedulerAcceptAllCertificates	If true, all security certificates (including not valid ones) will be accepted when accessing the scheduler.aspx page via a secured protocol. The default value is *false*.	`<add key=" CMSSchedulerAcceptAllCertificates" value="true" />`
CMSSchedulerUserName	User name under which scheduler.aspx should be accessed (e.g. when using windows authentication). The default value is "" (blank username).	`<add key=" CMSSchedulerUserName" value="office\myname" />`
CMSSchedulerPassword	Password for the user name under which	`<add key="`

	scheduler.aspx should be accessed. The default value is "" (blank password).	CMSSchedulerPassword" value="mypassword123" />

Special settings for security

By adding the following keys to your web.config, you can configure certain Security options:

Key	Description	Sample Value
CMSCheckParameters	Indicates if parameter checking is allowed. Read this topic for more details. The default value is *false*.	<add key=" CMSCheckParameters" value="true"/>
CMSReportCheckParameters	If true, an exception reporting the parameters is thrown when the parameters do not match. Read this topic for more details. The default value is *false*.	<add key=" CMSReportCheckParameters " value="true"/>

Special settings for user interface cultures

By adding the following keys to your web.config, you can configure UI cultures:

Key	Description	Sample Value
CMSDefaultSpellCheckerCulture	Specifies the default culture of the built-in spell-checker. This culture is used when the dictionary for the currently selected content culture is not found. The default value is *en-us*.	<add key=" CMSDefaultSpellCheckerCulture" value="en-US" />
CMSShowLogonCultureSelector	Indicates if the user interface logon page should display a drop-down list with available user interface languages. The default value is *true*.	<add key=" CMSShowLogonCultureSelector" value="false" />
CMSDefaultUICulture	Specifies the default UI culture. If you use this key, you also need to rename the *~\CMSResources\CMS.resx* file to *CMS.en-us.resx* and the *CMS.en-nz.resx* file to *CMS.resx*. This is needed because the *CMS.resx* file is used when the *(default)* option is selected in users' *Preferred user interface culture*. The default value is *en-us*.	<add key=" CMSDefaultUICulture" value="en-nz"/>
CMSUseSQLResourceMana	Changes the priority of used localization	<add key="

gerAsPrimary	resource strings to: 1. custom.resx 2. cms.resx 3. database (Site Manager -> Development -> UI Cultures) The default value is *true*.	CMSUseSQLResourceManagerAsPrimary" value="false" />
UICulture	Specifies the default user interface culture. The default value is *en-us*.	<add key="UICulture" value="en-us"/>

Special settings for file synchronization on web farms

By adding the following keys to your web.config, you can enable or disable web farm synchronization of certain kind of files stored in the file system:

Key	Description	Sample Value
CMSWebFarmEnabled	Indicates if Web farms are enabled (true) or not (false). The default value is *false*.	<add key="CMSWebFarmEnabled" value="true"/>
CMSWebFarmServerName	Code name of the web farm server. This value is used for native web farm synchronization support. The default value is "" (empty string).	<add key="CMSWebFarmServerName" value="server1" />
CMSWebFarmSynchronizeFiles	General key determining if files should be synchronized (true) or not (false). This key enables synchronization of: • Attachments • Meta files • Media files • BizForm files • Avatars • Forum attachments The default value is *true*.	<add key="CMSWebFarmSynchronizeFiles" value="true" />
CMSWebFarmSynchronizeAttachments	Enables/disables synchronization of atachments. The default value is *true*.	<add key="CMSWebFarmSynchronizeAttachments" value="true" />
CMSWebFarmSynchronizeMetaFiles	Enables/disables synchronization of meta files. The default value is *true*.	<add key="CMSWebFarmSynchronizeMetaFiles" value="true" />
CMSWebFarmSynchronizeMediaFiles	Enables/disables synchronization of media files.	<add key="CMSWebFarmSynchronizeMed

	The default value is *true*.	iaFiles" value="true" />
CMSWebFarmSynchronizeBizFormFiles	Enables/disables synchronization of BizForm files. The default value is *true*.	<add key="CMSWebFarmSynchronizeBizFormFiles" value="true" />
CMSWebFarmSynchronizeAvatars	Enables/disables synchronization of Avatars. The default value is *true*.	<add key="CMSWebFarmSynchronizeAvatars" value="true" />
CMSWebFarmSynchronizeForumAttachments	Enables/disables synchronization of forum attachments. The default value is *true*.	<add key="CMSWebFarmSynchronizeForumAttachments" value="true" />
CMSWebFarmSynchronizeDeleteFiles	Enables/disables synchronization of deleted files. The default value is *true*.	<add key="CMSWebFarmSynchronizeDeleteFiles" value="true" />
CMSWebFarmMaxFileSize	If the **CMSWebFarmSynchronizeFiles** key is enabled, you can limit the maximal size of synchronized files using this key. The value is entered in kiloBytes and files larger than this value will not be synchronized. The default value is *2147483647 (int.MaxValue)*.	<add key="CMSWebFarmMaxFileSize" value="1024" />
CMSWebFarmUpdateWithinRequest	If true, web farm servers are updated with changes made on the other servers once per request. If false, web farm servers will be updated based on the Synchronize web farm changes scheduled task - recommended for high-traffic sites. The default value is *true*.	<add key="CMSWebFarmUpdateWithinRequest" value="false" />

Connection string

The database used by Kentico CMS engine is specified by the connection string CMSConnectionString in the **/configuration/connectionStrings** section. Here's an example of such connection string:

```
<add name="CMSConnectionString" connectionString="Persist Security Info=False;
database=CMS;server=myserver;user id=sa;password=mypassword123;Current
Language=English;Connection Timeout=120;" />
```

Index

- 3 -

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- K -

- L -

- M -

- N -

- O -

- P -

- T -

- U -

- V -

- W -

- X -

- Y -