# Kentico CMS Developer's Guide 4.0

# Table of Contents

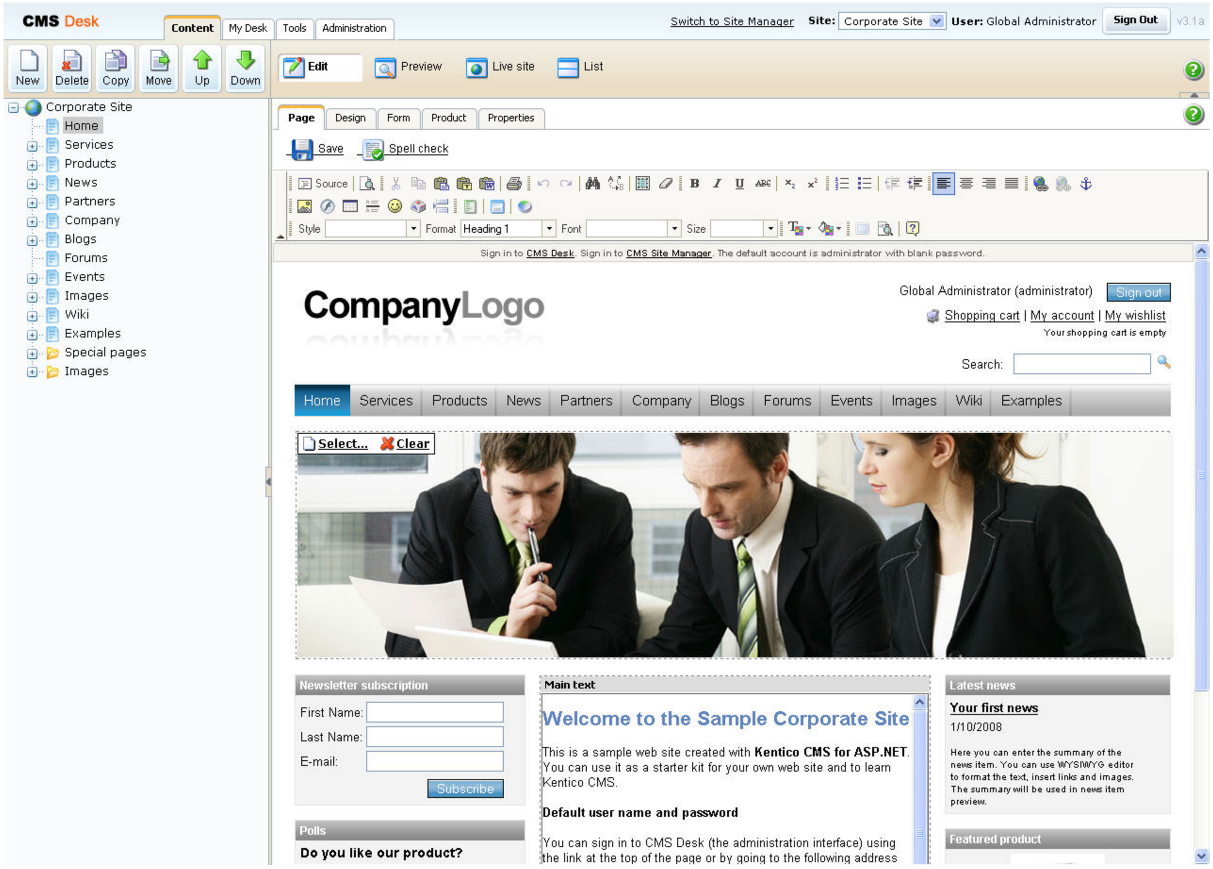
# Part I

# 1 Introduction

## 1.1 Introduction

Thank you for using Kentico CMS, the content management system for ASP.NET developers. Kentico CMS 4.0 empowers developers in creation of dynamic web sites. 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 web site.
- 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 Where do I get help?

If you need help, please visit www.kentico.com/support.aspx and www.kentico.com/DevNet.aspx where you can find:

- Support forums
- Knowledge base
- Technical articles

## 1.3 Version history

### January 16, 2009 - Version 4.0

**New:**

- New module: Categories
  - enables additional categorization/taxonomy of documents
- New module: Tags
  - enables tagging any documents (not only blog posts) with tags and displays a tag cloud
- New module: Alternative forms
  - enables creating multiple versions of editing form in BizForms, document types and custom tables
  - allows you to define custom forms, such as user registration form
- New module: Custom tables
  - enables creating custom tables in case you need to store some data or web content outside of the CMS content repository
  - it's useful if you need to work with large number of records in a single table
- New module: Banned IPs
  - allows administrators to limit access from chosen IP addresses
- New module: Bad words
  - allows administrators to forbid nasty words and replace them, report them or disable given user action, such as forum post or blog comment
- New module: Avatar gallery
  - allows administrators to create pre-defined avatars (user images) and let users choose their avatar or upload a new one
- New module: Time zones
  - allows you to display time and date calculated for the given web site or user
  - it's useful for global communities so that users can see e.g. forum post time based on their time zone
- New module: On-line users
  - allows you to display users who are currently on-line and kick them if they do not follow the site rules
- New module: Message boards
  - allows you to create message boards and comments for documents
  - the module supports moderation, subscriptions, rating and other features
- New module: Notifications
  - enables subscribing to content changes
  - provides a general notification framework that can be used not only for e-mail, but also for any custom notification gateway, such as instant messengers
- New module: Friends
  - allows site members to maintain relationships with other site members and share personal information with them
- New module: Report abuse

  - allows site visitors to report offending content, such as forum posts, blog comments and others
- New module: Groups
  - allows you to create social networking sites with user communities (groups)
  - allows inviting users to join the group
  - allows users to create group-specific content and applications, such as wiki-like pages, forums, message boards, media libraries and polls
- New module: Media library
  - enables easy management of media files, such as images, videos or music
  - the files can be accessed directly without any script and they are not stored in the database, so you can work also with large files
- Reviewed module: Forums
  - many new features including:
    - quoting
    - mark as answer
    - attachments
    - BB-style editor
    - BBCode support
    - customizable forum layouts
    - locked forums/threads
    - sticky threads
    - and many others
- Reviewed module: URL rewriting and aliases
  - you can now specify default alias path or URL redirection for every domain alias of the web site, which means that you can display different home page to visitors based on the domain name they use to access the site
  - many new features including:
    - support for unlimited URL Aliases for every document
    - tracking of marketing campaigns when some page is visited using some particular URL
    - custom extensions of pages and uploaded files
    - support for macros in URLs
- Reviewed module: Membership
  - improved member management
  - the user registration form can be easily customized with your own fields without programming
  - user registration may optionally require e-mail confirmation
  - administrator can optionally choose that all new user registrations must be approved first
  - you can now send mass e-mails to multiple users, roles or groups
  - improved user interface for management of large number of users with filtering and alphabetical sorter
  - support for mixed mode authentication
  - you can now use both Forms and Active Directory authentication together
- Administration UI: Extended functionality of MyDesk -> My profile.
- Administration UI: Asynchronous document deletion and moving.
- Administration UI: New information splash screen on logon dialog.
- Administration UI: New version of the WYSIWYG editor.
- Administration UI: Extended functionality in the web part zone properties.
- Administration UI: Extended functionality of System tables.
- Administration UI: Overviews where the document type, page template, product and category are used.
- Blogs: Bulk Approve/Reject comments functionality in MyDesk and improved comments management in Tools.
- Blogs: Support for blog moderators.
- Blogs: Support for blog post subscriptions.
- Blogs: Added support for tagging, categories and content rating

- Blogs: On-site editing is now supported for site members without access to CMS Desk
- Blogs: You can now ping other blogs and get trackbacks to your posts
- Controls: New format of inline controls macro now supports named parameters.
- Controls: New controls - context menu, error messages, datasources, filter, BBEditor, ...
- Controls: Minor improvements in localization controls from ExtendedControls.
- ECommerce: New web parts - N newest products, product filter
- ECommerce: Multiple default tax classes for department.
- ECommerce: Document types can be configured as product types which have enhanced functionality.
- E-mail engine: Support for custom subject, plain text, from, Cc and Bcc e-mail addresses in e-mail templates.
- E-mail engine: Enables more robust sending of e-mails. All e-mails from the system are sent through the e-mail queue and if sending fails, they are kept in the queue and can be sent later. It also allows to archive sent e-mails so that you can easily find out if some e-mail was sent or not.
- Event Log: The object changes can now be now logged.
- Event log: Easier browsing of events using Previous and Next buttons.
- General: New contexts for easier access to the data of current objects.
- General: Logon page now checks the CapsLock status.
- General: Minor improvements in file import module.
- General: All the obsolete methods from previous versions have been removed.
- General: Performance optimizations regarding memory consumption, amount of data selected from the database and ViewState.
- General: Google site map support
- Import/Export: Minor improvements in import/export.
- Messaging: Minor improvements in messaging.
- Content: New image editor for the attachments.
- Content: Content rating support. Allows site visitors to rate news, articles and any other content.
- Content: Image editor for the attachments. Allows content editors to resize uploaded images, convert them to another format, trim to chosen size and convert to grayscale.
- Newsletters: Performance improvements for large number of subscribers.
- Newsletters: Customizable subject in (un)subscribe templates.
- Output filter: Minor improvements in output filtering.
- Portal engine: On-click AJAX configuration of the web parts. Most web parts now support AJAX on one click. You only need to check the box Use update panel in the web part properties and the web part uses AJAX rather than classic postbacks.
- Reporting: Minor improvements in reporting.
- Security: New default (generic) roles.
- Security: Windows Live ID support.
- Settings: Improved management of settings.
- Staging: Minor improvements in staging.
- System: Administrators can now receive e-mail notification when some error is added to the event log.
- User contributions: Minor improvements in user contributions.
- Web analytics: Statistics from specific period of time can now be deleted.
- Web analytics: User registration statistics can now be logged.
- Web farms: Support for restarting of all the servers.
- Web parts: Improvement of many existing web parts.
- Web parts: News datasource and viewer webparts.

**Changes:**

- Administration UI: All pages are now based on master pages.
- Administration UI: Administration UI now uses CSS classes instead of the SkinIDs.

- General: Changes in filesystem and architecture to support separable modules in future versions.

**Bug fixes:**

- Administration: Fixes in RTL languages layout.
- Administration: Minor fixes in administration UI.
- BizForms: Image button wasn't set as default button of the form when custom layout was used.
- Bizforms: CSS stylesheet didn't work when bizform's layout was edited.
- BizForms: CAPTCHA control didn't work when the BizForm is used as a inline control.
- BizForms: Notification e-mail wasn't sent if recipients e-mail adresses ended with ';'.
- BizForms: 'UseColonBehindLabel' didn't worked on ASPX templates after postback.
- BizForms: Incorect links of uploaded files when a bizform from one site was used on another site.
- Blogs: URLs in blog comments were incorrect when inserted without 'http://'.
- Blogs: Minor fixes in blog comments.
- CMSDesk: Context menu wasn't displayed well if content tree was large and scrolled down.
- Content: Editable texts in PagePlaceholder with default page template didn't work at Page tab.
- Content: Images didn't appear occasionally during heavy load when output cache was on.
- Content: Content of PageTemplates properties -> Webparts couldn't be saved.
- Content: Moving document to another site caused an error when document name and alias had been changed before.
- Content: Document could be archived by user without required permissions.
- Content: Some links and image URLs weren't resolved on ASPX sample web sites site when output filter was switched off.
- Content: URLs of images from another site were incorrenct when using multiple web sites on a single domain (subfolders).
- Content: ID of the document custom data couldn't be used as an alias.
- Content: Documents restored after deletion had different NodeGUIDs than the deleted version.
- Content: CMSMenu tooltip didn't display special characters properly.
- Content: Template 'Text with side column' caused problems to editable regions in IE7.
- Controls: Images in submenu items didn't have the "alt" atribute.
- Controls: BasicTabControl caused index out of bounds exception with specific configurations.
- Documentation: Minor fixes in the documentation.
- ECommerce: Order status history wasn't up-to-date when the status was changed by custom payment provider.
- ECommerce: Minor fixes in permission checking in administration UI.
- E-mails: Minor fixes in resolving template based e-mails.
- File import: Minor fixes in the file import.
- Form controls: ECommerce form controls didn't work properly with property UseCodeNameForSelection set to true.
- Forums: Minor fixes in posts approval.
- Forums: 'Subscribe to post' link wasn't displayed when Maximum nesting level was set to 1.
- General: Title tag wasn't as a first tag in the head section of the page.
- General: Minor fixes in working with the flash files.
- General: Field category created in FieldEditor couldn't contain apostrophes.
- General: Custom security handler didn't work correctly with windows authentication.
- General: 'Exclude documents from search' setting didn't work if the value ended with ';'.
- General: Decimal numbers weren't displayed correctly for different cultures.
- General: Data from custom fields weren't updated in the application cache.
- General: Custom modules didn't appear in CMSDesk -> Tools if 'Display' permission for the module didn't exist.
- General: Custom data handler caused issues when working with document types and when retrieving files via GetFile.aspx.
- General: Some controls didn't generate valid html code.

- General: Alias paths weren't unique if documents had names starting with the same long prefix.
- General: Some controls were missing in VisualStudio toolbox after adding a reference to Kentico CMS API libraries.
- General: URL rewriting didn't worked properly if extensionless and SSL were used together.
- General: Pages inherited templates only from published pages which was causing inconsistency in design.
- General: Default connection string was always used when using GeneralConnection.
- General: ValidationHelper.GetDouble method didn't reflect current culture settings.
- General: SignOut button didn't work with cookieless settings.
- General: User role couldn't change permissions even if the role had the permission to do so.
- General: PageInfo didn't contain proper DocumentPublishedVersionHistoryID.
- Import/Export: Only documents of default culture version were imported to the site which had been imported before.
- Import/Export: Importing data with attachment might cause errors.
- Import/Export: Too long queries were generated during export, causing errors.
- Import/Export: Imported/exported site's cultures weren't preselected correctly.
- Newsletters: CSS stylesheet wasn't applied after uploading an attachment when editing newsletter template.
- Newsletters: 'Send now to all subscribers' option didn't work when editing dynamic newsletter.
- Polls: Two different polls couldn't coexist on the same page when the caching was on.
- Scheduler: SqlDateTime overflow error appeared when date wasn't in allowed interval.
- Scheduler: Scheduled tasks didn't run when using multiple web sites on a single domain (under subfolders).
- Staging: Staging could work even if 'Staging service authentication' settings didn't match on source and target servers.
- Staging: Root element was missing when synchronizing object's subtree.
- Staging: Automatically created blog months didn't appear in the staging task list.
- Staging: Synchronization tasks weren't logged correctly when the workflow was on.
- Web parts: Grid webpart didn't display data of two document types which had fields of the same names.
- Web parts: Minor fixes in Content slider webpart.
- Web parts: Minor fixes in Flash webpart.
- Web parts: Minor fixes in Tree view menu webpart.
- Web parts: Site map redirected to '#' when javascript action of the menu item was used.
- Web parts: Document pager webpart didn't have the 'Check permission' property.
- Web parts: Minor fixes in QueryDataList webpart.
- Web parts: Minor fixes in Lightbox webpart.
- Web parts: Minor fixes in Google maps webpart.
- Workflow: There was an incorrect message when document was submitted to approval.
- Workflow: Form fields weren't validated when a document was checked-in.
- Unigrid: Wrong item was deleted when someone else changed item order at the same moment (order status, workflow step, ...).

## August 6, 2008 - Version 3.1a

**New:**

- Administration UI: HTML editors are used for editing invoice and e-mail templates.
- Administration UI: Link to an existing document can be created from the context menu.
- Administration UI: New macro selector control is used to insert all possible macros to the text.
- Administration UI: Color picker control now includes preview of the selected color.
- Administration UI: Section summary for the pages of administration UI.

- Administration UI: Minor improvements in all UI.
- General: Improved output filter functionality.
- General: All regular expressions for validation and macro resolving were optimized.
- General: Performance optimizations for amount of data and memory consumption.
- General: Configurable selection between standard and session authentication cookies.
- Ecommerce: Product price can be optionally displayed with or without tax.
- Ecommerce: Product option list can be now paged.
- Ecommerce: User account can be created at the same time as customer account from CMSDesk.
- Ecommerce: All currency related values can be now recalculated when changing main currency.
- Ecommerce: Tax class details can be edited when selecting tax classes for specified product.
- Ecommerce: Shopping cart process step indexes can be now optionally hidden or visible.
- Ecommerce: My Account web part tabs can be now displayed either in horizontal or vertical layout.
- Ecommerce: Shipping options are now displayed with their charge in the checkout process.
- Ecommerce: All errors while loading payment provider, payment form or checkout process step are now logged.
- Event log: Event details can be exported to the file.
- Event log: Events can be optionally logged to the file now.
- Modules: Module description field was added.
- Modules: Module permission description field was added.
- Modules: New Display permission for all modules. Module is now hidden in tools menu when user is not granted with module display permission.
- Newsletter: Imported users can be optionally subscribed / unsubscribed to the selected newsletters.
- Newsletter: Email queue can be now deleted at once.
- Scheduler: User can see how many times each task was executed.
- Settings: Settings can be exported to the file.
- Staging: Reject task is now generated for synchronization with target server when document is rejected.
- Staging: Update doc task is now generated for synchronization with target server when document is moved up or down in the content tree.
- System: Testing e-mails can be sent with attachments.
- Web farms: Web farm synchronization tasks can be now filtered by server.
- Web farms: Servers can be now disabled to avoid tasks logging for them.
- Web farms: MachineName field was added to the web farm tasks.
- Web parts: Improved scrolling text web part.
- Web parts: Improved current user web part.

**Changes:**

- General: XHTML output filter is now turned off by default on all sample web sites.
- Web parts: Properties NestedRepeaterID and NestedDataListID were replaced by NestedControlsID property.

**Bug fixes:**

- Administration UI: RTL languages layout was completely reviewed and fixed.
- Administration UI: Some pages weren't properly handled for cross-site scripting.
- Administration UI: Minor fixes for IE7 and FF3.
- Administration UI: Security fixes for some UI pages
- BizForms: Incorrect security check was applied in permission dialog.
- BizForms: Administration of BizForm layout used too strict permissions.
- Booking module: Event allowed negative capacity.
- CMSDesk: Select file dialog couldn't upload file on Windows Vista.
- CMSDesk: Document links with workflow could cause inconsistency in document tree if the

document was moved.
- CMSDesk: Notify when leaving page wasn't applied in some cases.
- CMSDesk: Spell checker didn't work for words with some culture-specific characters.
- CMSDesk: Published information in document properties wasn't evaluated properly.
- CMSDesk: The document wasn't inserted on the correct position when the default order was changed from alphabetical to first.
- Content: Metadata content wasn't properly encoded to the output HTML.
- Content: When the image caching was on, original image could be returned instead of the resized image.
- Content: Cached attachment wasn't removed from the cache if the attachment was updated from API.
- Content: ASPX editable controls were losing the content with specific workflow operations.
- Content: Copying of published document didn't include the published attachments.
- Content: Rollback to the oldest version in version history could lose the version attachments.
- Content: File field could lose the attachment when copying the culture version from specific language.
- Content: When the attachment field of the document type was removed, attachments weren't deleted.
- Content: Allowed extensions for upload fields didn't have the same format in all settings.
- Content: Uploaded files allowed "+" sign in the filename, which couldn't be correctly processed by IIS7.
- Content: GetFile script didn't handle the permissions correctly.
- Controls: Editable image could cause null reference exception in specific scenarios.
- Controls: Radiobutton list form control didn't apply some settings correctly.
- Controls: Form control text area didn't validate the field length properly.
- Controls: Property UseImagePath for editable image control didn't work properly.
- Controls: Radiobutton list form control couldn't have empty default value.
- Controls: Change password control didn't handle the Visible property correctly.
- Documentation: Minor fixes in the documentation.
- ECommerce: Payment methods and Shipping options weren't displayed properly in their selection.
- ECommerce: Available amount validation could not work correctly in specific scenarios.
- E-mails: Some e-mails contained multiple Content-Type headers.
- E-mails: GetEmailTemplates method threw an exception.
- File import: Files couldn't be imported to the location covered by workflow process.
- General: Automatic import of Windows authenticated users wasn't working correctly for non-existing users.
- General: Changing the code name of the web site could cause problems with live site if the application wasn't restarted.
- General: Application couldn't properly start in Integrated mode of IIS7.
- General: Culture of the live site wasn't changed correctly in some cases with full page caching configuration.
- General: Uzbek cultures weren't defined properly.
- General: Changing the order of items didn't clear the cached objects properly.
- General: Localized transformations weren't cached.
- General: Float number validation and conversion couldn't handle small numbers with exponent.
- General: Param tag of the object element could cause incorrect HTML code when processed by output filter.
- Import/Export: Forum posts weren't properly imported from old version package.
- Import/Export: Custom web part categories couldn't be handled by import process in some specific cases.
- Import/Export: Some delete tasks couldn't be processed correctly on import.
- Import/Export: Existing site was deleted if import fails because of the existing site with the same name.

- Import/Export: Process log couldn't work in specific scenarios.
- Import/Export: Preselection of objects wasn't working properly when repeated.
- Import/Export: Next button wasn't disabled in FF3.
- Import/Export: Document fields with only field length changed weren't updated on import.
- Newsletter: Some e-mails could use different culture for unsubscription link and for other newsletter content.
- Newsletter: Newsletter module couldn't handle large amount of e-mails in specific scenarios.
- Newsletter: Some e-mails were duplicated in specific scenarios.
- Newsletter: Resend functionality wasn't working properly.
- Newsletter: Scheduled task for the newsletter wasn't rescheduled if configuration changed.
- Newsletter: New issue couldn't be placed to the Newsletter archive
- Newsletter: Generation of the e-mail queue was using too many database queries.
- Newsletter: Some newsletter pages didn't check all required permissions.
- Newsletter: Newsletter archive didn't resolve the macros properly.
- Output filter: Button text containing tag characters could damage the output HTML code.
- Polls: Multiple polls couldn't work properly on the same page.
- Polls: Incorrect security check was applied in permission dialog.
- Portal engine: Some pages could be displayed with wrong page template in heavy load.
- Portal engine: If the CSS stylesheet of the site wasn't set, the page generated an empty stylesheet tag.
- Sample web sites: Minor fixes in design and HTML code of the page templates and stylesheets.
- Scheduler: Some tasks could run two times under specific conditions.
- SiteManager: Some Administration pages could be accessed from CMSDesk in SiteManager context.
- SiteManager: User couldn't be deleted when he was allowed to be forum moderator.
- SiteManager: Site culture couldn't be changed in license editions without multilingual support.
- SiteManager: Localized transformation name couldn't be entered in transformation properties.
- SiteManager: Document type editing pages didn't have the same length.
- Staging: Document task didn't contain product information when the product was created from the new document dialog.
- URL Rewriting: Some URLs weren't resolved correctly for cookieless configuration.
- URL Rewriting: Messaging pages weren't properly excluded from URL rewriting and were logged as not existing content pages.
- URL Rewriting: Page wasn't properly redirected in specific cases with FF browser.
- Web analytics: Log processing couldn't handle large amount of data at once.
- Web analytics: Returning visitors were logged only on the second visit.
- Web farms: Too many web farm tasks or too much web farm task data could kill the application on it's start.
- Web parts: Clone web part couldn't work with some specific web part settings.
- Web parts: Content slider wasn't working properly if the delay time was 0.
- Web parts: Google maps web part couldn't handle API key for both domains with and without www.
- Web parts: Query repeater web part didn't use the Item separator property correctly.
- Web parts: User contribution web part couldn't resolve the path properties.
- Web parts: Language selector didn't display culture names in ASPX page templates.
- Web parts: Date and time web part couldn't handle some format expressions.
- Web parts: Contact and ignore list didn't use the Zero rows text property properly.
- Web parts: Video web part wasn't working properly in FF3.

## May 29, 2008 - Version 3.1

**New:**

- New module: The new Messaging module allows users to send private messages.
- CMSDesk: BizForms can now be exported as single objects.
- Content staging: Content staging supports staging of metadata objects.
- Web parts: New web part ForumUnsubscription.
- Web parts: New messaging web parts (Inbox, Outbox, Send message, Contact list, Ignore list).
- Web parts: New web part Send to friend.
- Web parts: New web part Random products.
- Web parts: New web part Top N products by sales.
- Sample websites: New E-commerce site template presenting the E-commerce module capabilities.
- Sample websites: Corporate site now contains an example of a printing page.
- E-commerce: Newly added support for product options.
- E-commerce: Products can now be filtered by site.
- E-commerce: Order item price and name can be edited in existing orders.
- E-commerce: The Organization ID/Tax registration ID can be optionally required.
- E-commerce: User can now create document product when creating a new document.
- Import/export: Import/export now supports incremental deployment.
- Import/export: New, more user friendly user interface.
- General: Culture CSS class is now included in page body class.
- General: Kentico CMS now supports shortcut icons for web sites.
- Web analytics: Web analytics module now supports repeated visits using IP address filtering. Users with disabled cookies can also be logged.

**Changes:**

- CMSDesk: Both Menu items and standard documents now share the view mode in CMSDesk.
- Administration interface: More listings with many data records now support data filtering.
- Import/Export: Export of single object is now asynchronous.
- Content staging: Document page template is now synchronized together with the document.
- Web parts: General code review and minor improvements of all existing web parts.
- Documentation: New E-commerce guide.
- Documentation: New Users guide.
- Documentation: New web parts overview.
- E-commerce: Printer style for invoices now includes header and footer information for all pages.
- Forums: Improved usability of macros.
- Setup: Setup process is now asynchronous.
- Newsletters: Improved usability of macros.
- Workflow: Improved usability of macros for workflow emails.
- General: Full page caching now reflects the browser type.
- General: Kentico CMS can now handle upload and download of large files (up to 2GB) saved in file system.
- Performance: Major improvements in GetFile script performance and minor optimizations in system processing.
- Performance: Memory optimizations.

**Bug fixes:**

- BizForms: Export of larger amount of BizForm data to MS Excel 2007 didn't work correctly.
- BizForms: E-mail attachments weren't handled correctly.
- CMSDesk: File import tool didn't import files with names containing some special characters.

- CMSDesk: File import tool didn't save mime types correctly.
- CMSDesk: List view didn't allow deletion of all documents after filtering.
- CMSDesk: Sorting in the List view didn't work correctly.
- CMSDesk: My Documents didn't display linked documents correctly.
- CMSDesk: Settings 'Content Management New document order' didn't work correctly.
- CMSDesk: Permissions weren't checked correctly when creating a new page.
- CMSDesk: Update of a linked document damaged the document's name path.
- CMSDesk: User couldn't create link to an uploaded file with apostroph in it's name.
- CMSSiteManager: Email templates didn't reflect the site settings.
- Content staging: Deletion of the linked documents wasn't logged.
- Content staging: Deletion of document didn't delete Publish task for this document.
- Controls: CMSTabControl generated wrong where conditions for database queries.
- Controls: CMSTreeview didn't reflect the Expand/Collapse images property.
- E-commerce: PayPal and Authorize.NET didn't work for cultures using comma as decimal separator.
- E-commerce: Currency couldn't handle wrong format strings.
- E-commerce: ECommerce module didn't synchronize correctly on Web farms.
- E-commerce: Discount level wasn't applied correctly when displaying price details.
- E-commerce: Deletion of some objects didn't handle the dependent objects correctly.
- Booking system: Editing event attendee didn't allow email address change.
- Booking system: Exported iCalendar file couldn't be opened in MS Outlook 2003.
- Form user controls: Clear button in Select user form control didn't work correctly.
- Form user controls: International number and Query selector didn't reflect the Allow empty value property of field.
- Form user cotnrols: Country selector didn't work correctly.
- Form engine: If validation upon regular expression failed error message was displayed for all fields using regular expression validation.
- Form engine: Validation upon regular expression was missing for some form controls.
- Form engine: Fields of type boolean didn't work correctly with values true/false and 0/1.
- Form engine: DateTime fields weren't validated properly.
- Forums: Ad-hoc forum wasn't created when user subscribed before the first thread was inserted.
- Forums: Forum search displayed previously deleted posts.
- Forums: Forum base URL wasn't saved when creating a new forum.
- Forums: Forum breadcrumbs displayed differently in different view modes.
- Forums: Deletion of forum posts didn't work correctly.
- Newsletters: Newsletter issues were always shown in archive.
- Newsletters: Newsletters with large amount of subscribers weren't sent correctly when application restarted or application had multiple instances.
- Newsletters: Scheduled task wasn't deleted when dynamic newsletter was deleted.
- Newsletters: Newsletters weren't always sent in site default culture.
- Newsletters: Anchor links in newsletters weren't resolved correctly.
- Portal engine: DOCTYPE page definition wasn't applied properly.
- Portal engine: PageInfo didn't handle the IsPublished property correctly.
- Web analytics: Campaign logging didn't work for redirected documents.
- web analytics: Logging file downloads didn't work for files redirected to disk.
- web analytics: Log processing wasn't thread-safe for multiple application instances.
- Web analytics: Pie charts displayed only items with value over 5%.
- Web analytics: The order of days might be wrong at the turn of the months.
- Web farms: Web farm synchronization tasks weren't deleted properly.
- Workflow: Copying documents of container type using workflow was failing.
- Workflow: Workflow email wasn't sent after the Publish step.
- Workflow: Workflow emails weren't always sent.
- Workflow: When the published document was rejected, it was still published on the live site.

- Workflow: In some cases, documents couldn't be restored from the recycle bin after the import process.
- Workflow: When publishing a document, the message 'Are you sure you want to navigate away from this page?' wasn't always shown correctly.
- Webparts: Web parts didn't reflect the Show for document types property in Design mode view.
- Webparts: Some viewer web parts didn't hide when the inner control was hidden.
- Webparts: CMSMenu didn't render correctly when a submenu item had a Menu item image set.
- Webparts: CMSListMenu didn't render correctly when submenu indicator property was set.
- Webparts: Static text web part didn't resolve inline macros.
- Webparts: Related documents didn't work correctly for nonpublished documents.
- Webparts: Property Top N didn't work correctly when Check permission property was set.
- Webparts: Properties didn't handle values dependent on culture settings correctly.
- Webparts: Users couldn't select document type transformation from different site.
- Webparts: The Item separator property of the Lightbox web part could not be edited.
- Webparts: The MyAccount web part didn't allow hiding of all its tabs.
- Webparts: Property Use server time didn't work correctly for DateTime web part.
- Webparts: The Search results web part didn't reflect the Query string key property.
- Webparts: The Complete search dialog web part didn't keep the filter information for ASPX templates.
- Webparts: When the SearchDialog web part was placed after the SearchResults web part on page it didn't work correctly.
- Webparts: Setting filter for specific culture sometimes caused hiding the New document link for User contributions list web part.
- Webparts: The User contributions list web part displayed New document link for anonymous user in a special case.
- Webparts: Web part files weren't created when the web part was cloned.
- Webparts: Web part CMSMenu didn't reflect the Render image alt property.
- Webparts: Web part SiteMap didn't reflect the Render link title property.
- Webparts: The property Generate indentation inside link for TreeMenu web part didn't work correctly.
- Webparts: The property Submenu indicator image for web part TreeMenu caused incorrect rendering.
- Webparts: The Paged text web part didn't use the Path selector for Path property.
- Webparts: The Paged text web part didn't use Site selector for SiteName property.
- Webparts: The Paged text web part didn't reflect the 'Pager CSS class' property.
- Webparts: In special cases the Calendar web part threw an exception when other month should have been displayed.
- Webparts: The Forum tree web part for ad-hoc forums didn't reflect the Expanded tree property.
- Webparts: The Forum tree web part didn't reflect the Post path property.
- Webparts: The Similar products by sale web part didn't work correctly if property Document types was set.
- Webparts: The Document viewer web part didn't reflect the Select top N documents property.
- Webparts: The Forum search results web part didn't reflect the On site management property.
- Webparts: The ShoppingCartMiniPreview control wasn't rendered correctly in ASPx templates.
- Webparts: the Query datalist control didn't reflect the ShowFirstLast property in the ASPX templates.
- Webparts: The DateTime web part didn't reflect the Date and Time format property correctly in ASPX templates.
- Webparts: The option Top & Bottom was misssing for Pager position property of Document pager web part.
- Webparts: In some special cases the Query data grid web part threw exception when Display as link property was used.
- Webparts: The Edit contribution web part used only Simple path selector instead of Path selector.
- Webparts: For some web part setting both Stylesheet theme and Skin ID properties caused an error.

- Webparts: Multiple Tree menu web parts on one single page didn't work correctly.
- webparts: The SkinID property wasn't applied correctly for web part Logon form.
- Webparts: The Related documents web part didn't have the Order by and Where condition properties.
- General: ASPX page templates from older versions didn't have correct page title.
- General: Settings Combine files with default culture didn't apply correctly.
- General: The page template header tags weren't used in some cases.
- General: Users or search engine crawlers could get an error page when the browser didn't set the User-agent property.
- General: In some cases, the resource strings weren't properly loaded when the application started.
- General: Multiple linked documents could cause errors when Filter out duplicate documents was enabled in viewer controls.
- General: GetFile didn't reflect document-level permissions and published from/to properties.
- General: In some cases users were logged out when browser didn't keep session information in a new frame or window.
- General: Too many WYSIWYG editors on the page could cause the browser to crash or throw a javascript error.
- General: Data macros that weren't resolved because of missing data were resolved to their expression instead of an empty string.
- General: Permissions for network folders weren't checked properly.
- General: Deletion of some objects didn't delete associated meta files.
- General: Resolving URL filter damaged AJAX response if the response contained URLs starting with ~/.
- General: For extensionless configuration the URL rewriter wasn't able to handle URLs with query strings containing a semicolon.
- General: ASPX extension wasn't correctly applied to the page URL in special cases and extensionless configuration.

## January 29, 2008 - Version 3.0

**New:**

- Compatibility: Added support for Visual Studio 2008 and .NET Framework 3.5 (you can use Visual Studio 2005 as well).
- Compatibility: Added support for Windows Vista, including IIS 7.0 configuration.
- Compatibility: Added support for Mac OS browsers - you can now use both web site and administration interface with Safari 3.0 and Firefox 2.5 on Mac.
- New module: The new GeoMapping module allows you to display content on Google maps. You can use it to display your offices, stores, partners, etc.
- New module: Event Manager for managing events and their attendees.
- BizForms: The notification e-mail may contain uploaded documents as an attachment.
- BizForms: The notification e-mail may use a custom layout.
- BizForms: The autoresponder now supports macro expressions.
- BizForms: You can restrict access to administration of particular forms.
- BizForms: You can use an image for the submit button and you can define the position of the submit button in the custom form layout.
- Controls and web parts: More settings were added to the pager configuration in Repeater, Datalist and other controls/web parts.
- Controls and web parts: The navigation controls now have a new property Word wrap that allows you to specify if word wrapping is allowed.
- CMS Desk: You can specify starting path of the user in the content tree in CMS Desk and restrict user from browsing the complete content tree.

- CMS Desk: New Design mode interface with drag-and-drop support.
- CMS Desk: Context menu support in content tree.
- CMS Desk: FCKeditor was updated to version 2.5.
- CMS Desk: The documents that point to some URL are now displayed with an arrow in the content tree to show that the document is a redirect.
- CMS Desk: New permission Design web site allows local users (without global administration authorization) modify the design of page templates.
- CMS Desk: The list view now supports filtering by name.
- CMS Desk: You can customize the behavior of the web application and administration interface and make minor changes to the administration interface using the new customization features described in Developer's Guide -> API programming and Kentico CMS internals -> Customizing the administration interface and web application events.
- Content staging: You can configure automatic synchronization of all content to the live server in a scheduled interval.
- E-commerce: Tracking number field added.
- E-commerce: Added support for custom fields in customer profile, order, SKU.
- E-commerce: The discount levels can be restricted to particular departments.
- E-commerce: Sales reports were added.
- E-commerce: The shopping cart optionally displays price calculation (discount and tax summary).
- E-commerce: The payment URL now supports data macros in format {%fieldname%} that inserts any value of the shopping cart, current context  or customer.
- E-commerce: Added support for customer credit.
- E-commerce: New web part Similar products by sale for displaying box with "Customers who bought this product also bought...".
- File management: Improved performance of the GetFile.aspx script, added support for redirection to the disk version of the file.
- Forums: Added support for on-line editing and deleting of posts by forum administrators.
- Form controls: You can now choose which form controls should be available in report parameters designer.
- Form controls: The multiple choice and radio buttons controls now support vertical/horizontal alignment.
- General: System objects, such as templates, layouts and others may have a teaser image.
- General: Reports, form layouts and e-mail templates may contain images uploaded to the give object.
- General: Dialogs with Save button now support CTRL+S shortcut for saving the changes.
- General: The global event handler now supports OnBeforeInsertNewCultureVersion and OnAfterInsertNewCultureVersion events.
- General: The CSS styles can now depend on browser type by specifying styles in format .IE6 . MyCssClass.
- General: New macro {&<path>&} allows you to specify path or its parts in the Where condition of web parts.
- General: You can define your own custom macros and use them throughout the system.
- Import/export: You can easily export and import a single object, such as page template, web part, etc.
- Multi-lingual support: The content tree in CMS Desk displays an icon for documents that are not translated to the currently selected language.
- Multi-site support: You can configure multiple web sites to run on a single domain, in different subfolders.
- Newsletters: You can clone newsletters and newsletter templates.
- Newsletters: You can add custom fields to the subscriber profile.
- Newsletters: The subscription dialog allows you to subscribe to several newsletters at once.
- Newsletters: New My subscriptions web part allows users to manage their subscribed newsletters.
- Newsletters: New Newsletter archive web part allows you to publish previous newsletter issues on your web site.

- Newsletters: It's now possible to choose if you want to send the confirmation e-mails when the user subscribes/unsubscribes.
- Newsletters: The newsletter issue is now sent to subscribers immediately, without waiting for the next scheduler cycle.
- Newsletters: You can add attachments to the newsletter and insert in-line images that are sent with e-mail.
- Localization: You can add custom resource strings to the custom.resx file and override standard strings.
- Performance: Several performance improvements were made to page processing and image displaying.
- Relationships: It's now possible to add custom fields and custom (XML) data to the relationships (the feature requires custom-developed user interface).
- Reports: You can clone reports.
- Security: The page content access can be globally configured so that the permissions are checked for all pages, none of them or only for pages within secured site areas.
- Web analytics: New reports "Countries" and "Browser types" were added.
- Web analytics: It's possible to choose particular reports and disable others to save space and performance.
- Web parts: The documentation is now part of the web part properties
- Web parts: The transformation name and culture code properties may contain a macro expression (e.g. {%classname%}.preview).
- Web parts: New web part Language selection.
- Web parts: A new web part Lightbox allows you to display content in a new layer and browse documents without postbacks.
- Web parts: New WMP Video, YouTube, Google Video, Real Media and Quicktime web parts allow you to easily integrate videos into the page
- Web parts: New Flash web part allows you to easily integrate Flash animations.
- Web parts: New Content slider web part allows you to create a slide show of documents.
- Web parts: New Scrolling text web parts allows you to scroll documents, such as latest news.
- Web parts: You can easily clone web parts, including their code for further customization or extension.
- Web parts: You can create custom web part layouts, which allows you to fully customize the design of web parts.
- Workflow: The non-published pages may throw 404 Page not found if the Site Settings -> Web Site -> Page not found for non-published documents option is turned on.

**Changes:**

- CMS Desk: The Code tab is now only displayed if the CMSShowWebPartCodeTab value in the web.config is set to true.
- CMS Desk: The Code tab is now only displayed if the CMSShowWebPartBindingTab value in the web.config is set to true.
- CMS Desk: The Custom fields tab is no longer displayed when custom fields are not defined.
- CMS Desk: The default installation contains only spell checkers for en-us and en-gb cultures to save space. You need to download additional dictionaries from our site as described in Developer's Guide -> Content Management -> Using the built-in spell-checker.
- CMS Desk: The Recent documents dialog now displays only the latest 50 documents sorted in a descending order.
- E-commerce: The wishlist is now connected with particular site.
- Forums: The e-mail is not automatically pre-filled into the textbox in the new post dialog if the user is signed in.
- Polls: The cookie indicating if the user voted in the poll now uses a codename rather then ID.

**Bug fixes:**

- General: Next Run time didn't change after editing Start time of Scheduled task.
- General: Only last item was saved to NodeCustomData document property.
- General: Some form controls didn't work within user contributions.
- General: Culture didn't change when swithing to edit the document from MyDesk.
- General: Root document wasn't shown in Documents waiting for my approval.
- General: SpellChecker didn't recognize some words with specific suffixes.
- General: Site deletion wasn't logged in Event log.
- General: EditableItems weren't able to store CDATA sections in <![CDATA[Content]]> format.
- General: Content staging didn't support linked documents.
- General: No attachments were copied when creating new culture version of cms.file document using workflow.
- General: E-mail addresses didn't allow '+' and '-' characters.
- General: Culture of the document was determined from URL path if URL path and Alias path were same.
- General: Linked attachment could have point to different culture version than requested in specific situations.
- General: UserInfo wasn't correctly configured to be serializable and caused problems with Out-of-process session storage.
- General: Content attribute of meta tags was missing when the value was empty.
- General: Non-existing users weren't authenticated on first page request with Windows authentication.
- General: /getdoc/… links were not resolved properly with specific multilingual conditions.
- General: Preferred culture code was not validated against the allowed site cultures.
- General: License information page was not properly loading styles and images with extension less configuration.
- Web parts/controls: UseFlatView property of ForumGroup web part didn't work properly.
- Web parts/controls: It wasn't possible to set Path, CultureCode and SiteName properties in EditContribution web part.
- Web parts/controls: Application path was included in image path of EditableImage web part that caused problems when the site was imported to the root.
- Web parts/controls: Image tag was generated even if no image was selected in EditableImage web part.
- Web parts/controls: Random document web part didn't work correctly when caching was on.
- Web parts/controls: Sorting wasn't preserved while changing pages in CMSGrid.
- Web parts/controls: Pager of CMSDataList and CMSRepeater caused exception when these controls were nested.
- Web parts/controls: ClassNames selector didn't allow choosing more items if these had nearly same code names.
- Web parts/controls: MinLength property of CMSEditableRegion didn't work.
- Web parts/controls: CMSRepeater didn't preserve ClientID of controls in transformations after post back.
- Web parts/controls: Breadcrumbs' Show current item property always returned true.
- Web parts/controls: Breadcrumbs couldn't display more document types at once.
- Web parts/controls: Language selector didn't work when extension-less URLs were used.
- Web parts/controls: Some web parts didn't support ReloadData method.
- Web parts/controls: CMSRepeater and CMSDataList didn't work correctly when displaying more document types.
- Web parts/controls: Registration web part was not generating SHA1 passwords properly.
- Web parts/controls: Web part properties editor allowed selecting of the file field which is not supported on the web part properties form.
- Web parts/controls: ForumSearchResults web part didn't use Unicode for searching, problems with Cyrillic alphabet, etc.

- Web parts/controls: Random document web part didn't support multiple document types.
- Web parts/controls: CMSPagePlaceholder rendered additional <div> element in Preview mode.
- Web parts/controls: CMSWebPartZone renders additional <div> in preview mode.
- Web parts/controls: New item edit mode button was displayed just for Menu item document types.
- CMSDesk: MaxTreeNodes of dialogs with content tree contained only 1000 documents per parent node hiding the other documents.
- CMSDesk: Frame resizer didn't work correctly for RTL languages.
- Installation: All licenses were deleted when installing to existing DB without creating new objects.
- CMSDesk: Relationship to the manually entered document path could not been added.
- CMSDesk: CTRL+S shortcut wasn't working properly with certain E-commerce pages.
- CMSDesk: Certain edit mode controls were not using UI culture.
- CMSDesk: New link page limited the number of displayed documents in tree to MaxTreeNodes.
- CMSDesk: Version history and Workflow tabs used workflow information from incorrect document culture.
- CMSDesk: Large files stored in the database too big for the Connection to process could have not been deleted.
- CMSDesk: Documents without version history were deleted into the owner's recycle bin instead of the recycle bin of the person who deleted them.
- CMSDesk: Destroying of the latest document version may have caused exception when working with the document further.
- CMSDesk: Grid with filtering could hide filter form in certain situations if no results were found.
- CMSDesk: Document could not be deleted by its owner after it was submitted to approval.
- CMSDesk: File import procedure didn't properly check if the target document allowed CMS.File as child document.
- CMSDesk: Tasks configured just for one execution configured to be deleted may have not been
- called.
- CMSDesk: Two concurrent validation messages were displayed when both regular expression and length validation were configured in the document form.
- CMSDesk: When deleting the document, published version was deleted to the recycle bin instead of the latest version.
- CMSSiteManager: Under specific conditions, global template could have been retrieved instead of the local template.
- CMSSiteManager: Code name fields were not validated in certain locations.
- CMSSiteManager: Task configuration did not allow tasks running only on weekend days.
- CMSSiteManager: Page template tree was not correctly refreshed after deleting the page template.
- Newsletters: Unsubscribe link wasn't translated to different languages.
- Newsletters: Sender name wasn't used in confirmation e-mails.
- Newsletters: Absolute links in newsletters weren't correctly resolved if SSL was used to access CMSDesk.
- Newsletters: Localization macros didn't work in (un)subscription letters.
- Newsletters: Subscriber list caused Javascript errors in Newsletter preview.
- Newsletters: Update of Sent emails field was not thread-safe, in some cases wrong value could be written.
- Content staging: Documents order was changed when new document was synchronized.
- Content staging: No content staging task was created when the document was archived.
- Forums: When there were more forum groups on a page of ASPX website, then the forum groups didn't hide after choosing some forum.
- Forums: Certain Unicode characters weren't supported in the forum post subject.
- BizForms: Localization macros didn't work in auto-responder e-mail body.
- BizForms: Exception occurred when entered default value was longer than max length in field editor.
- Web analytics: Period settings were lost when switching between reports.
- Web analytics: Month selection was not disabled when selecting the year statistics.
- Web analytics: Stored procedures contained [dbo]. which caused problems on specific

configurations.
- Content staging: Documents with attachments and workflow may have been synchronized incorrectly with specific configurations.
- Polls: Poll web part was creating duplicit items on specific postbacks causing exception.
- Polls: Polls controls HTML code wasn't XHTML valid.
- Polls: Only global administrator was able to add new polls.
- Import/Export: Import and export wasn't able to process sites with many attachments stored in the database, ended up with OutOfMemory exception.

## October 8, 2007 - Version 2.3a (Service Release)

**Bug fixes:**

- Performance: Improved performance and optimized queries for large number of documents.
- In-line controls: The in-line controls didn't work correctly in the ASPX page templates.
- Controls: Template data pages displayed an invalid number of pages.
- Controls: CMSDocumentValue tag didn't display correct content when publish from/to was used.
- Controls: Image selector doesn't work correctly after postback.
- Import/export: The import didn't work for web sites that used workflow assigned to all document types in the given scop.
- Import/export: The attachments were not imported correctly in some special cases.
- E-commerce: The strings were not localizable in the fifth step of the shopping cart.
- URL rewriting: The parameters containing a dot (.) were not processed correctly with extension-less URLs.
- Search: The search query for custom document types without custom fields was incorrectly generated.

## September 17, 2007 - Version 2.3

**New:**

- New module: Blogs
- New module: Polls
- New module: Reporting
- New module: Web analytics
- New module: User contributions
- Documentation: Enhanced documentation, more API examples.
- Linked documents: You can create links to existing documents so that they are displayed in multiple sections/categories.
- Installation: Simplified support for medium trust environment.
- Installation: It's now possible to deploy the web site to a local folder for later copying to the server over FTP.
- BizForms: It's now possible to specify if you want to display colons (:) in field labels.
- BizForms: The visitor who submits the form may now receive an automatic personalized e-mail response.
- CMS Desk: It's now possible to choose the language of the administration interface in the logon screen.
- CMS Desk: New report "Outdated documents" allows users to see document older than specified number of days.
- CMS Desk: The document owner property has been added to the documents. The property allows you to assign responsible users to the documents and display their documents in My Desk -> My

Documents.

- CMS Desk: The newly created document can be optionally added as first, last or at its alphabetical position. You can configure the ordering in Site Manager -> Settings -> Content management -> New document order.
- CMS Desk: The Properties tab and its sub-sections can be optionally hidden from chosen roles. It allows you to customize and simplify the look of the editing interface. You need to enable personalization in Site Manager -> Settings -> Content management -> Personalize user interface by permissions. Then, you can customize the user interface by configuring permissions for module CMS User Interface in CMS Desk -> Administration -> Permissions.
- CMS Desk: It's possible to specify a different CSS stylesheet for every editable region.
- Web parts/controls: The CMSTreeMenu control/Tree menu web part now support mouse-over images.
- Web parts/controls: The CMSTreeMenu, CMSTabControl, CMSSiteMap, CMSBreadcrumbs and CMSListMenu controls now allow you to supply custom DataSource. They also allow you to get/set their rendered HTML code.
- Data engine: It's now possible to write your own data provider.
- E-commerce: Wishlist support has been added.
- E-commerce: The payment options are now dependent on shipping options.
- E-commerce: The order list now displays the orders in different colors based on their status and the order can be moved to the previous/next status using the action buttons.
- E-commerce: The tax may be optionally zero if Tax ID is entered.
- E-commerce: It's now possible to edit user departments in the User properties dialog.
- E-commerce: Support for discount level has been added - every customer can have a global discount on all products.
- E-commerce: Support for volume discounts has been added.
- E-commerce: It's now possible to specify taxes based on states, not only on countries.
- E-commerce: It's now possible to specify the shopping cart and wishlist URL in the Site Manager -> Site Settings -> E-commerce section.
- E-commerce: A document type can be configured so that whenever a new document of that type is created, a new product is also automatically created and the data is copied to the product.
- E-commerce: It's now possible to track the number of products in stock that can be ordered and optionally limit orders to number of available items.
- Forums: The moderator can approve a post including its replies (sub-posts).
- Forums: The user may be required to enter a security (CAPTCHA) code when adding a new post to the forum. This feature helps you avoid spam.
- Import/Export: It's not possible to import only new items without overwriting duplicate items.
- Site Management: The site deletion is now asynchronous to avoid timeouts when deleting a large web site.
- Newsletters: You can now import and export subscribers.
- Newsletters: You can specify the CSS styles for every newsletter template, separately from the template body.
- Newsletters: Improved user interface of the newsletter issue editor.
- Newsletters: More robust e-mail sending. The newsletter has been successfully tested with 1000 subscribers.
- Performance: Improved performance of file retrieval and optimization of some queries.
- Search: Improved performance
- Search: The search page now uses URL parameters instead of postback so that it's possible to use the Back browser button on the linked pages.
- Security event handler: Added new method OnFilterDataSetByPermissions.

**Changes:**

- CMS Desk: When a new document is created, its CSS stylesheet is set to "inherit" by default.
- CMS Desk: The CSS stylesheet property has been moved to Properties -> General and it's no

longer versioned.
- CMS Desk: The destroy document and destroy version history options are no longer offered to users without Destroy permission.
- CMS Desk: The delete all culture versions option is no longer offered on single-language web sites.
- CMS Desk: The content tree is now displayed in chosen language when using multilingual support.
- Controls examples: The CMSControlsExamples folder content is now available as a part of the Corporate Site sample web site, in the Examples section.

**Bug fixes:**

- BizForms: The record ID wasn't available in the OnAfterSave event.
- BizForms: The textboxes in the "Send form data to e-mail" section were not validated.
- Sites: The page might end with request time out when deleting large web sites.
- Newsletters: The editable regions might not be displayed correctly when editing a newsletter issue text.
- Newsletters: The sender e-mail format wasn't validated when creating a new newsletter.
- Newsletters: The permissions were not tested correctly on the Subscriber properties dialog.
- Newsletters: The links in newsletters were not correctly made absolute when using https protocol.
- E-commerce: The disabled products could be added to the cart.
- E-commerce: The user who created the order was not stored in the order history.
- E-commerce: Invalid formatting of the forgotten password dialog.
- E-commerce: The e-mail addresses were not validated when editing order properties.
- E-commerce: When the product image was removed in the Form dialog, it hasn't been removed in the Product dialog.
- E-commerce: Some methods of the custom e-commerce providers were not called.
- Web parts/controls: The Select only published property of the Repeater web part didn't work correctly.
- Web parts/controls: The Paged text web part didn't work correctly with QueryString paging mode.
- Web parts/controls: The Related documents web part didn't work correctly in ASPX page templates.
- Web parts/controls: The Tab menu web part didn't display the first item.
- Web parts/controls: The search results didn't reflect the "combine with default culture" option.
- Web parts/controls: The logon.ascx web part displayed an error in Visual Studio, when an ASPX page template was displayed in design mode.
- Web parts/controls: The Repeater web part set the Check permissions value incorrectly to true when it was added to the page.
- Web parts/controls: The container title wasn't localizable.
- Web parts/controls: some strings were not localized in the Bindings dialog in the Web part properties.
- Web parts/controls: The HTML code and special characters displayed in the alt tag weren't encoded in the menu controls.
- Form engine: The labels contained invalid LabelFor ID for user controls.
- Import/Export: The sites exported in the previous versions couldn't be imported in the 2.2 web site.
- Import/Export: The document permissions were not imported if they were defined by a user that doesn't exist in the new web site.
- Search: The search raised an exception if the root document wasn't published.
- Forums: The validation didn't work correctly when editing existing forum posts.
- Forums: The emoticons were not displayed correctly when editing existing forum posts.
- Forums: The WYSIWYG editor didn't work when editing existing forum posts.
- Forums: The post statistics included the non-approved items.
- Forums: The notification e-mails contained invalid URL when the post was added from the administration interface.
- Multilingual web sites: The language selection didn't work correctly for pages that use URL Path.
- Multilingual web sites: The default culture was not applied correctly based on the current domain.
- CMS Desk: The design mode didn't reflect the chosen UI culture.

- CMS Desk: The user e-mail wasn't validated in My Desk -> My profile and in Administration -> Users.
- CMS Desk: It was possible to copy/move page into sub-page which caused errors and inconsistencies in the database.
- CMS Desk: The content tree displayed the code name of the web site instead of display name.
- CMS Desk: Some of the extended characters used in the content were encoded as HTML entities which caused that the strings were not found by the full-text search engine.
- URLs: The = character was added to the forbidden characters that are replaced in URL.
- URLs: The links created in the WYSIWYG editor didn't work correctly with "combine with default culture" option turned on.
- Workflow: The notification e-mail contained workflow step code name instead of display name.
- Workflow: The documents might not be visible on the web site if they were archived and the re-published.
- Workflow: The documents using workflow didn't work correctly when they were copied to another web site.
- Workflow: The documents using workflow didn't work correctly when restored from the recycle bin.
- Workflow: The modification of page template caused that the document workflow step was set to Edit.
- Workflow: The "New workflow step" dialog contained a non-translated validation error.
- Workflow: The notification e-mail didn't point to the correct language version of the document when using multilingual support.
- Document types: The icon wasn't renamed when the document type code name was changed.
- Document types: The default values with apostrophe caused problems.
- Document types: The default value range wasn't validated correctly.
- WYSIWYG editor: The image wasn't preserved in FireFox when the <P> elements were removed around the <IMG> element.
- New site wizard: The Delete button didn't work.
- Site Manager: The e-mails were not validated correctly in the Site Manager -> Administration -> System -> E-mails dialog.
- Caching: The Cache minutes property in web parts didn't override the global settings.

## June 8, 2007 - Version 2.2

**New:**

- General: Kentico CMS now fully supports Medium Trust environment and compiled web projects. See Developer's Guide -> Installation and deployment -> Additional configuration tasks -> Configuration for Medium trust environment. You need to save all virtual objects to the file system before compilation using CMSSiteManager -> Administration -> System -> Deployment by clicking the button “Save all virtual objects to disk”.
- General: Page templates of the portal engine now can have custom content for the HEAD element specified.
- General: The CSS stylesheet, page title, description and keywords can now be set for every section of the web site and inherited to underlying documents.
- General: The title format is now configurable in Site Manager -> Settings -> Page title format and allows you to use any common document field in the title (e.g. the document name).
- General: You can now specify which sections of the web site should be accessible only over the HTTPS protocol. The system then automatically redirects the user to the HTTPS protocol if the user tries to access the page over HTTP. You can configure this in the CMS Desk -> Content -> Properties -> Security. You need to have SSL configured correctly on your web server.
- General: Any document type can now behave as page (menu item) - this can be configured in the document type properties.
- General: Most of the web parts now work correctly with Enter key.

- General: The extended characters in aliases (URLs) are now automatically converted to basic latin characters. This doesn't apply to complex scripts, such as Chinese or Arabic.
- General: The users now have two new fields with account creation date and last successful logon.
- General: It's now possible to use URLs based on alias path rather than on the GUID. You can configure this in Site Manager -> Settings -> URLs -> Use permanent URLs (false means use alias path).
- CMS Desk: Spellchecking support was added.
- CMS Desk: You can now upload files in the file selection dialog which simplifies inserting images into the text in the WYSIWYG editor.
- CMS Desk: You can now easily create multiple documents of the same type in the same section of the web site using the "Save and another" button.
- CMS Desk: The list view mode now allows you to delete/destroy multiple documents at once.
- CMS Desk: The editing interface in the Page and Form tab may optionally notify the user when she is trying to leave the page without saving the changes. You can configure this in Site Manager -> Settings -> Content Management -> Confirm content changes.
- CMS Desk: The content tree now displays the + sign only for items with sub-items.
- CMS Desk: The transformations can now be edited directly from the web part properties.
- CMS Site Manager: New diagnosing tools for testing e-mails and file system permissions were added to the Site Manager -> Administration -> System dialog.
- Controls: New Paged text web part now allows to display long text on multiple pages.
- Controls: The navigation controls and web parts now have the RenderImageAlt/Render image alt property that allows you to specify if alt attribute should be rendered for images used in the menu items.
- Controls: The Random document web part now supports the WHERE condition property.
- Controls: The Breadcrumbs control/web part now displays the menu items while respecting particular settings of every item (such as show in navigation).
- Controls: The Breadcrumbs control/web part now supports RTL languages.
- Controls: The image gallery web part now supports moving to the previous, next and chosen image.
- Controls: The CMSEditableRegion control used in the ASPX templates now has a property InheritContent that allows you to inherit the content of the editable regions on sub-pages placed under some page (menu item).
- Controls: New HEAD HTML Code web part allows you to render code into the HEAD element of the page.
- Controls: New web part/control Editable image/CMSEditableImage allows you to specify regions for images similar to editable regions.
- Controls: All properties of web parts now support the localization strings in format {$key$} where key is the string key name specified in Site Manager -> Development -> UI Cultures -> Strings.
- Controls: All web parts now have a property Display to roles that allows you to specify which roles should see the given web part.
- Controls: The CMSTreeMenu control/Tree menu web part now supports mouseover style settings.
- BizForms: You can now dynamically specify the values in the From, To, Subject, Display text and Redirect to URL fields using the macro expressions like {%firstname%} where firstname is the name of the field that will be merged with the text.
- BizForms: You can now specify the subject of the notification e-mail.
- BizForms: The system now automatically renames the BizForm and its database table if you import a BizForm of the same name with another site.
- BizForms: The country/province selection dialog now stores the values as code names rather than IDs.
- BizForms: It's now possible to export data to Excel 2007.
- Forums: The users can now have their picture and signature and edit them using the My Profile web part.
- Forums: The users can now specify their nickname in the My Profile web part.
- Forums: You can now configure the use of WYSIWYG editor for post editing.
- Forums: The users are now informed when they submit a post to a moderated forum.

- E-mails: The e-mails can be sent over SSL-secured SMTP server - you can set it up in Site Manager -> Settings -> E-mails -> Use SSL.
- Multilingual support: It's now possible to configure automatic use of default language version of the file if it's not translated - this option is now independent on the "Combine with default culture" option that is used for content. It can be configured in Site Manager -> Settings -> Web Site -> Combine files with default culture.
- Newsletter: The subscribers can have custom fields defined in Site Manager -> Development -> System tables.
- Workflow: It's now possible to configure the workflow scope for all document types.

**Bug fixes:**

- Controls: The CMSTreeMenu and CMSListMenu didn't recognize the selected menu item correctly if the URL path used a different prefix than the alias path.
- Controls: The PageManager didn't check the permissions correctly.
- CMS Desk: The scheduling dialog didn't work correctly with some cultures.
- Workflow: In very specific cases, the user could approve the document without being authorized to do that.
- CMS Desk: The virtual objects (layouts, CSS stylesheets) were not updated in the memory when the user modified them using the check-in/check-out function.
- CMS Desk: The browse tree permission didn't allow editors to browse content unless they had also the edit permission.
- CMS Desk: The WYSIWYG editor didn't work correctly with https URLs.
- CMS Desk: The default values of the custom fields were not applied when creating a new user.
- CMS Desk: The maximum number of tree nodes didn't work correctly with multilingual web sites.
- Controls: The MyProfile web part didn't work correctly on ASPX page templates.
- Controls: The Registration form web part didn't allow for empty value in the "Assign to roles" property.
- Controls: The related documents settings didn't work correctly in some web parts.
- BizForms: The custom query for retrieving drop-down list items couldn't use < and > characters.

**Changes:**

- Editor: The WYSIWYG editor is now placed in an IFRAME by default. If you experience any issue with design in the editing mode, please make sure you have <add key="CMSUseIFrameForHTMLEditorToolbar" value="true" /> in your web.config, in the appSettings section.
- Database: All tables now have the Enforce for replication property set to No.
- CMS Desk: The Approve/Publish button now saves the changes made to the text.
- E-commerce: When a customer is disabled, her user account is disabled as well.

## May 11, 2007 - Version 2.1e - Service Release

**Bug fixes:**

- Allowed extensions for uploaded files were not applied correctly.
- The extension of the postback URL was incorrect when using an extension different to .aspx.
- Missing scrollbars in several dialogs.
- Enter did not work as a default submit button in the search dialog.
- The PageInfoProvider didn't work correctly with addresses that were in format xxx.domain.com/xxx.
- The image gallery web part didn't work correctly in ASPX templates.
- The site import wizard didn't check if there's any package to be imported.
- The COM_CurrencyExchangeRate table contained duplicated records after export/import.

- The hashtables with queries and transformations were not updated when a document type was deleted.
- The site map web part/CMSSiteMap control didn't work correctly with URL redirection.
- The custom error message of the "user control" field and drop-down list field wasn't used.
- The values of the user custom fields were deleted when the user account general properties were updated.
- The BizForm property "Send form data to e-mail" couldn't be unchecked.
- The drop-down menu/CSS menu didn't work correctly with mouse-over CSS class.
- The BizForm web part didn't work correctly with custom code in the Code tab.
- The move of the document to another site didn't work correctly.
- The tab menu/CMSTabControl didn't work correctly with URL redirection.
- The passwords in plain text format were not case sensitive.
- The GetFileUrl functions did not resolve the links correctly.
- The SMTP server name now supports an explicit port number in format smtpserver:port.
- Improved quality of resized images and client caching of images.
- Improved and more stable site import process.
- The version history of attachments was not exported/imported.
- Sorting in CMSDataGrid control/Grid web part didn't work correctly.
- The rollback didn't preserve correct links to attached files.
- The import process didn't update the workflow step correctly in the imported documents.
- The BizForm module didn't display a scrollbar in the Data -> Select fields dialog.
- The content staging synchronization didn't work correctly in special cases when the first synchronization task was "Published".
- The content staging synchronization didn't work correctly for attachments in multilingual documents.
- The information on the current workflow step wasn't updated in some cases.
- The Random Document web part didn't work correctly in ASPX templates.
- The search results were not correct for the "all words" option.
- The paths to CSS files in the Forums and E-commerce modules did not work correctly for extension-less URLs.
- The BizForm Data dialog now supports paging.
- The controls displaying relationships might fail in some cases due to invalid query.
- The custom layouts and master page template were not updated in the cache when modified.
- The root document was displayed instead of the default page in some cases, even if the user viewed the live site.
- The import process didn't import the ASPX templates, VB.NET inline user controls correctly.
- The document name path wasn't updated correctly when copying or moving a document in some cases.
- The Shopping Cart content was deleted before the OnFinishOrder method was called.
- The Shopping Cart didn't use the custom method GetPaymentURL.
- The web part properties dialog didn't work correctly when the page alias path contained extended characters.
- The file upload didn't work correctly if the files folder was outside the web site and the web site root didn't have modify permissions assigned.
- The workflow didn't work correctly after it was disabled for the documents. Now the documents finish their workflow process and then they start working as a document without workflow.
- The non-approved content changes appear on the live web site when the document was being edited.
- The ViewState didn't work correctly in ASCX user controls inserted through the "General/User control" web part.
- The upload field in BizForms raised an exception when trying to update an existing record.
- The pager used in repeater and datalist kept invalid URL parameters used on other page/by other users.
- The web part configuration dialog wasn't displayed correctly when the alias path of the page

contained extended characters.
- The non-aproved images were not displayed when inserted into the page.
- The images in version history might not be displayed in some cases due to the loss of user session.

**Changes:**

- The BizForm e-mails can now be sent to multiple e-mail addresses separated with a semicolon (;).

**New:**

- The transformations can now be configured for different language versions - you only need to define the transformation with name in format transformation_<culturecode>.
- The menu controls can now load related table content if you specify a single document type, which allows you to filter the items using the WHERE expression.
- The CMSListMenu control/CSS List Menu web part now has a new property RenderLinkTitle/Render link title property that allows you to configure the control so that it displays a title for every link for better accessibility.

## March 20, 2007 - Version 2.1d - Service Release

**Bug fixes:**

- The permission "Check-in any document" was not applied.
- It was possible to approve a checked-out document.
- The unsubscribe link in the newsletter module contained invalid URL in case the web site was running in the site root.
- The field name couldn't be renamed.
- E-mails sent from the BizForms module did not support localized strings.
- The SkinID property of the search dialog could cause errors.
- Some of the session variables didn't support out-of-process session storage.
- The newsletter web parts didn't support localization of custom captions.
- The Add new button was displayed in the Detail view.
- The generated search query was invalid for container document types.
- The SQL Import displayed confusing messages when the connection string was not set.
- The deletion of all items in an order in the E-commerce module caused an exception.
- The RoleInfoProvider didn't raise the CustomDataHandler events.
- The shipping, billing and currency can now be localized using the localization strings.
- The layouts, transformations and web parts might be cached even after they were modified.

**Changes:**

- The FCKeditor shared toolbar can now be placed in a separate frame to avoid issues with design. You can set up the placement in a separate frame using the following web.config parameter: <add key="CMSUseFrameForHTMLEditorToolbar" value="true" />
  Please note: This value is now intended for testing purposes only, it will be fully supported in the next versions.

**New:**

- The BizForms data view now supports paging.
- All menu controls now support the custom menu item settings, such as styles or custom images. The mouseover action is supported only by CMSMenu control/Drop down menu web part.
- Added browser configuration for correct behavior of the web site when it's accessed by W3C

validator.
- The system now logs the restarts of the application and displays the time from the last restart.
- It's now possible to send support requests with system information from the Site Manager -> Support section.

## March 1, 2007 - Version 2.1c - Service Release

**Bug fixes:**

- The workflow notification e-mail was sent with a confusing subject to the person who was authorized to approve the document.
- The subscription in the forum didn't work correctly in the tree view mode - the user was subscribed to the whole forum instead of particular post.
- The Design tab was not displayed for the pages with inherited template.
- The Calendar web part didn't work correctly when placed on an ASPX page template.
- If the page contained multiple forum groups, they were not displayed when the user selected a forum in some forum group.
- The code placed in the Code tab of the web part didn't work correctly on sub-pages.
- The code placed in the Code tab of the web part didn't work for editable regions.
- When the site code name was changed, the system still didn't allow to create a new site using the original code name.
- When the site code name was changed, the images were no longer visible.
- The copying of file documents didn't copy the file.
- Fixed bugs in caching of layouts, transformations and web parts.
- The string resources didn't work correctly if stored in database.
- It wasn't possible to have two BizForms of the same code name on two different sites.
- The postback didn't work with extension-less URLs on Windows Server 2003.

## February 22, 2007 - Version 2.1b - Service Release

**Bug fixes:**

- Important security fix: The administration section didn't control the access rights correctly. It's highly recommended that you upgrade your system to 2.1b!
- The search dialog didn't work in the ASPX templates.
- The Current User web part didn't use the user name label CSS style.
- The CSS list menu rendered invalid code for inactive menu items.
- The product access permissions were fixed in the Content -> Product dialog.
- The image gallery web part didn't work correctly in the ASPX templates.
- The CMSTreeMenu control rendered invalid code in case some menu item image was specified.
- The page template code name uniqueness wasn't checked.
- The check-out function for the page layout didn't work correctly in some cases.
- The e-mails sent by the Newsletter module didn't display the line breaks in some e-mail clients correctly.
- The filtering in Customers and Orders lists didn't work correctly.
- The allowed file extensions that can be uploaded were case sensitive.
- The Relationship name property of web parts wasn't saved and applied.
- The ASPX page template files were not exported during the Export objects wizard.
- The Sign out button wasn't localized.
- Several bugs were fixed in the import/export process.

**What's new:**

- Partial translations were added: French, Italian, German, Spanish, Dutch, Czech.

## February 7, 2007 - Version 2.1a - Service Release

**Bug fixes:**

- The file name couldn't be entered for ASPX page templates.
- Document check-out could lead to a database dead-lock if the version history length was too small.
- Several bug fixes in the import/export procedures.
- The left tree menu didn't unfold when some document was opened using an e-mail notification link.
- The file upload didn't work correctly if files were configured to be stored in the file system.
- The ID of the web part added to a new page template wasn't unique.
- Missing scrollbars in the New document type wizard.
- The search engine caused an exception if multiple searches were performed at the same time.
- The cloned template modifications affected also the original template.
- The RSS feeds contained invalid news item URLs if running in the web site root.
- The password format was not set correctly when creating a new user.
- Paging didn't work correctly in the Image gallery web part.
- CMSRepeater and some other controls didn't work with custom ItemTemplate sections and other nested configuration sections.
- Portal page template inserted under the ASPX page templates didn't work correctly.
- The button Submit to approval might been still displayed after the document was published in some cases.
- After editing a page and pressing check-in the content controls were not locked until you navigated away from the page and back again.
- Language selector didn't work correctly.
- The portal engine-related tabs (dialogs) were not hidden for ASPX page templates.
- Several bug fixes in the Content staging module, including synchronization of move/copy operations.
- The visual inheritance didn't work correctly if the document inherited only from the root document in some cases.
- Added documentation of new database tables.

**Changes:**

- The CMS system resource strings were moved to the CMSResources/CMS.resx file which speeds up the installation, makes the translation easier and saves database storage space.

## January 30, 2007 - Version 2.1

**What's new:**

- Improved performance.
- New Forums module.
- New E-commerce module.
- New Content Staging module.
- Added full support for ASPX page templates and documentation.

- It's now possible to add custom fields to the user profile through the administration interface (Development -> System tables).
- The master page can be edited using a special tab "Master page" that is displayed for the root page. This allows you to edit DOCTYPE, HEAD element content, BODY element attributes and page layout of the master page.
- The document types can be marked as document types with "page template selection" so that the page template must be selected when a new document is being created.
- The Newsletter module now allows you to choose if you want to send confirmation e-mails to users when you subscribe/unsubscribe them through the administration interface.
- The virtual paths, such as "~/images/image1.gif", can be used also for the background attribute.
- The logon form now supports sending of forgotten passwords.
- The icons in content tree can now be clicked (for better usability).
- The Drop down menu (CMSMenu control) now supports separators.
- Support for nested repeaters/datalists.
- More robust import/export methods.
- The SQL Import utility now supports import of images.
- The web parts can now be placed directly on the ASPX pages or ASCX controls, outside the portal engine.
- New BizForm web part.
- New JavaScript web part.
- New My Profile web part.
- The Registration form web part now supports registration of users whose account needs to be activated by the administrator.
- New Web Services web parts (Datalist, Grid, Repeater) allow you to display content provided by a web service.
- New Master page tab in the CMS Desk allows you to easily modify the master page, including DOCTYPE, HEAD section and BODY attributes.
- New property Skin ID was added to several web parts.
- It's now possible to change the default culture of the web site through the user interface at any time.
- It's now possible to set content inheritance for every single page.
- It's now possible to create page templates with page template-specific HTML layout.
- The scheduled tasks can now be automatically deleted after they are completed.
- New menu action configuration section has been added to the Properties -> Menu dialog. You can use it to define an inactive menu item.
- New CMSDocumentValue control allows you to display the given value of the current document.
- The transformations can now use a new method IfEmpty to decide if the value (e.g. image field) is empty and display conditional text.
- New example of RSS feed with content from Kentico CMS was added to the Developer's Guide -> Extending modules -> RSS feeds.

**Changes:**

- The design mode is not applied only to the current page (page template), not on the whole page.
- The PDF documents are now returned with content disposition inline, so that they do not display a download dialog.

**Bug fixes:**

- The localization was not fully supported in the BizForm form.
- The CMSEditModeButtonAdd control rendered empty <span> element in the live site mode.

- The repeater/datalist/grid controls/web parts didn't work correctly if placed on the page template assigned to a document of other type than "page (menu item)".
- Page templates with empty parent category were not displayed in the page template selection dialog.
- Several dialogs were not displayed correctly in IE 6.0.
- The drop down menu was displayed as vertical when it was approved in the Preview mode.
- The date time picker might not work correctly when the content was in other culture than en-us.
- The "is custom string" box value in the UI Cultures section was not saved correctly.
- The "created by" field of the document was not set when a new document was created.
- The full-page caching didn't work correctly with personalized web sites (authenticated users).
- Document links couldn't be created correctly in the Static HTML web part properties in IE 7.0.
- Extended characters were not displayed correctly in page layouts and transformations.
- The List view mode was displaying only published documents.
- The file selection dialog did not work correctly for other file types than images.
- The output filter did not work correctly in some special cases.
- The e-mail form field didn't validate the e-mail addresses written in uppercase correctly.
- The BizForm module didn't support localization of all form captions.
- The Grid (CMSDataGrid), Datalist (CMSDataList) and Repeater (CMSRepeater) controls didn't work correctly if placed on page template assigned to non-page document type.
- The icons of the custom document types were not exported and imported.
- Storing passwords in hashed format didn't work correctly.
- The BizForm module did not save the uploaded files.
- The design mode of the web parts rendered an incorrect code of the table elements.
- The changes made to the web part container code were not updated on the live web site automatically.
- The session variables are now serializable, so that they can be used within a web farm with SQL Server session store.
- The files placed within a secured area didn't require authentication.

## November 29, 2006 - Version 2.0b

**What's new:**

- IE 7.0 and FF 2.0 are now fully tested with Kentico CMS and supported.
- It's now possible to get the time when the scheduler was last run and restart it manually if necessary.
- New Document page web part that can be used for displaying a list of documents, one document on every page. It's useful for paging of long articles and paging of "Take a tour" wizards.
- New Random document web part that enables displaying a random document, such as "featured product".
- Improved performance of the tree view in CMS Desk -> Content section.
- Improved performance of output filters and portal engine.
- New system of caching that enables caching of whole pages which improves the performance around ten-times in comparison to web part-level caching. The caching can be set up in document properties -> General.
- The information message about editing/preview mode is no longer displayed if the page is displayed inside the CMS Desk.
- Improved support for editing page layouts directly from the Design mode.
- TAB key is now supported in several code editors (such as page layout or transformation editor) throughout the user interface.
- The CSS list menu web part/CMSListeMenu control now supports ID prefixes for differentiating several list menus on the same page.

**Bug fixes:**

- Fixed design in master page template selection in the New site wizard.
- Fixed problems with scheduling engine that appears in some server environments where the server cannot request page from itself (due to firewall or DNS settings). If you experienced problems with crashing ASP.NET process, you need to add the following parameter to the appSettings section of your web.config file: <add key="CMSUseAutomaticScheduler" value="false" />.
- Datalist no longer renders a table element in the detail view.
- The web site now contains App_Browsers folder with configuration for search engines, such as Google, so that they can access sub-pages using friendly URLs (it workarounds a bug in ASP.NET 2.0 URL rewriting process).
- The forbidden characters were replaced with underscore (_) instead of configured replacement character in some cases.
- The newsletter module doesn't use the Sender from name when sending a newsletter issue that has been already mailed out.
- The newsletter module may display a scheduling dialog after saving the static newsletter configuration.
- The newsletter module doesn't resolve the e-mail address in <A> element correctly.
- The content tree may display a JavaScript error when the document name contains an apostrophe.
- The global administrator can select links or images from any web site even if he/she is not assigned to the web site.
- UI culture (string translations in BizForms, date/time format, etc.) is now set correctly on the web site based on the current user's preferences.
- More stable import/export module.
- String translation in Development -> UI Cultures -> Strings threw an error saying the string already exists.
- The page template wasn't editable in Design mode if visual inheritance was broken by the Inherit page levels property.
- The documents are now correctly displayed if they are placed directly under the page template with page placeholder that has a default page template specified.
- The web parts can now be correctly moved to newly added web part zones.
- The Breadcrumbs web part contained an extra <div> element.
- The files referred using their URL based on alias path did not reflect the currently selected content culture.
- The notification e-mail in BizForms module now contains correctly translated field names.
- The document creation through API failed for long names/alias paths.
- HTML files uploaded to Kentico CMS are now displayed directly in the browser, without displaying a download/open dialog.
- The search didn't work correctly for selected document types.
- The roles assigned to workflow steps were not exported/imported.
- The newsletter issue preview didn't display correct URLs when the site was running in the root.
- The user interface culture and translated strings were not displayed correctly in some cases.
- The content culture was set incorrectly when the session expired.
- The file selection dialog didn't work correctly and didn't contain the Clear button.

## November 7, 2006 - Version 2.0a

**What's new:**

- New design of the sample Corporate Site project.
- Improved performance and optimized output filters.
- The folder app_code/<sitename> is now automatically exported/imported with web site.

**Bug fixes:**

- If an invalid default alias path is used, the user is redirected to the root.
- Fixed bugs in the licensing mechanism.
- The connection string is now saved to the web.config file only after the database is successfully created.
- Documents with long names might not be saved.
- Import/export didn't save settings values correctly.
- The registration form raised validation on every postback.
- The web site might not be deleted.
- The URL path was incorrectly updated when alias path was modified.
- The WYSIWYG editor created links with domain, although they were inside the given site, the URL didn't follow the configured extension.
- The Save as new template button updated all pages using the original template.
- The breadcrumbs control displayed incorrect links.
- Several minor bug fixes.

## October 26, 2006 - Version 2.0

**What's new:**

**Framework and infrastructure changes:**
- All code rewritten to C# 2005, VB.NET version is no longer available, but development in VB.NET will be still supported.
- Visual Studio 2003/ASP.NET 1.1 is no longer supported.
- Improved and enhanced API. All operations are now accessible through the object model.
- Custom event handlers throughout the whole system that allow developers to enhance the code with custom actions when e.g. a new document is created, when a user is authenticated, etc.
- All possible settings (web.config settings, transformations, etc.) are now stored in the database instead of the file system.
- It’s possible to configure the web site so that it doesn’t write into the file system.

**New Portal Engine**
- It’s now possible to create pages using webparts in the web browser, without having to use Visual Studio.
- Several webparts are now available: navigation, content, membership etc.

**Multi-site Management**
- It’s now possible to run multiple web sites using a single database and code installation.
- The web sites can have different or shared users, content, structure and settings.

**Content Management**
- New database structure for storing content.
- Every site is “multilingual” by default, even if it uses only a single language. It means that it’s not necessary to configure multilingual support.
- Multilingual web sites may have different structures – it’s no longer necessary to have all documents in the default language.
- New, easier-to-use user interface with preview functionality (without staging) and simplified editing.
- It’s now possible to sort all content tree items manually using arrows instead of setting Menu item order value.
- All documents (not only pages) now can have their menu item caption, page title, keywords, order

and page template.
- All documents can be restored from the recycle bin after being deleted accidentally even without versioning.
- Workflow can be assigned to particular site sections and document types, instead of per-document type basis.
- Global search & replace.
- Improved link management using the permanent links that keep working after the target document has been moved or renamed.

**Security**
- Added membership provider implementation that enables use of Kentico CMS user management with standard ASP.NET 2.0 membership controls.
- Support for custom authentication (user and role information) from an external source.
- Support for automatic import of user account and roles from Active Directory/NT domain.
- Users can change their profile (e-mail, password, preferred culture) in the My Desk section.
- Customizable user profiles
- New user attributes "is editor" and "is global administrator".
- Local deny permission overrides the global permissions.
- Secured areas can be easily defined through the user interface.

**Form Engine**
- It’s possible to specify custom form layout for editing forms and BizForms.
- Custom validation (value range, regex) is now supported.
- Custom styles for every control.
- Support for field groups.

**Deployment**
- It’s possible to export the whole web site or its parts (e.g. document types) into a single file (on the development machine) and restore it on the target (production) server.
- Global search & replace enables easy updating of links.
- Kentico CMS now can run without disk write permissions and in the medium trust environment.

**File Management**
- All files are now stored in a special folder and retrieved using a dedicated ASPX page.
- It’s possible to store files only in the database.

**URL Rewriting**
- URL Rewriting now supports extension-less URLs without extra “dummy” files on the disk.
- Custom actions and URL rewriting rules are now supported.

**BizForms**
- Optional switching to advanced user interface with more options.
- Custom form layout, validation and styles.

**Newsletters**
- Completely new, more robust interface and API.
- Support for dynamic newsletters (such as last week's articles, etc.).

**Forums**
- Simplified, more flexible configuration.
- Forums can be optionally opened only for registered users.
- Support for ad-hoc forums.

**Scheduling**
- New scheduling sub-system enabling scheduling of custom tasks.

**Search**
- New provider-based model enabling use of custom search engine.
- Customizable search results design.
- Search results can be paged.

**CMS Controls**
- DataGrid, Repeater and DataList controls now support selection of only top N rows.
- DataGrid, Repeater and DataList controls now support paging.

# Part II

# 2 Kentico CMS Overview

## 2.1 Should I use Kentico CMS?

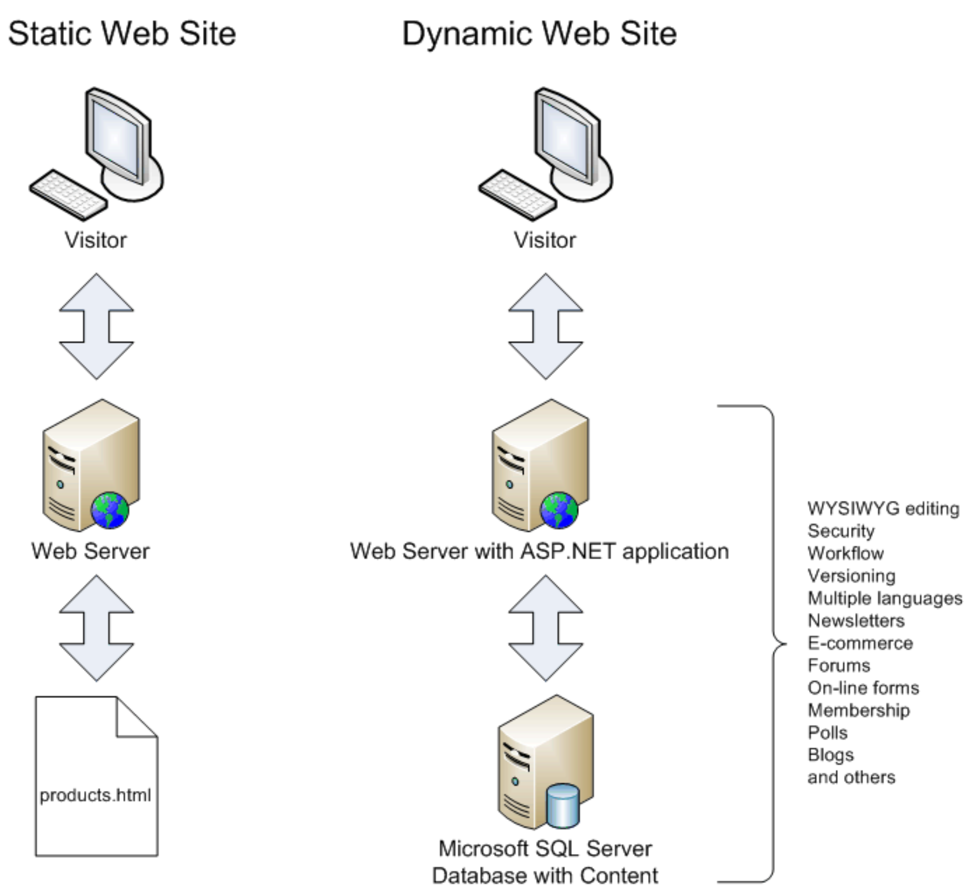
Before you dive into the details, it may be useful to understand the big picture. This chapter explains how the system works, describes its architecture and answers the most common questions you may have.

### 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 web sites. Unlike static web sites that uses static HTML files stored on the disk, a dynamic web site displays content from the database. Kentico CMS provides both content storage and all surrounding infrastructure to manage the content and display it on the web site. 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 web site 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** - web site 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 web sites. 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 web site 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 web site 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 web sites 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 web site structure, navigation, graphic design and integrate custom functionality**.

**Don't take our word for it**

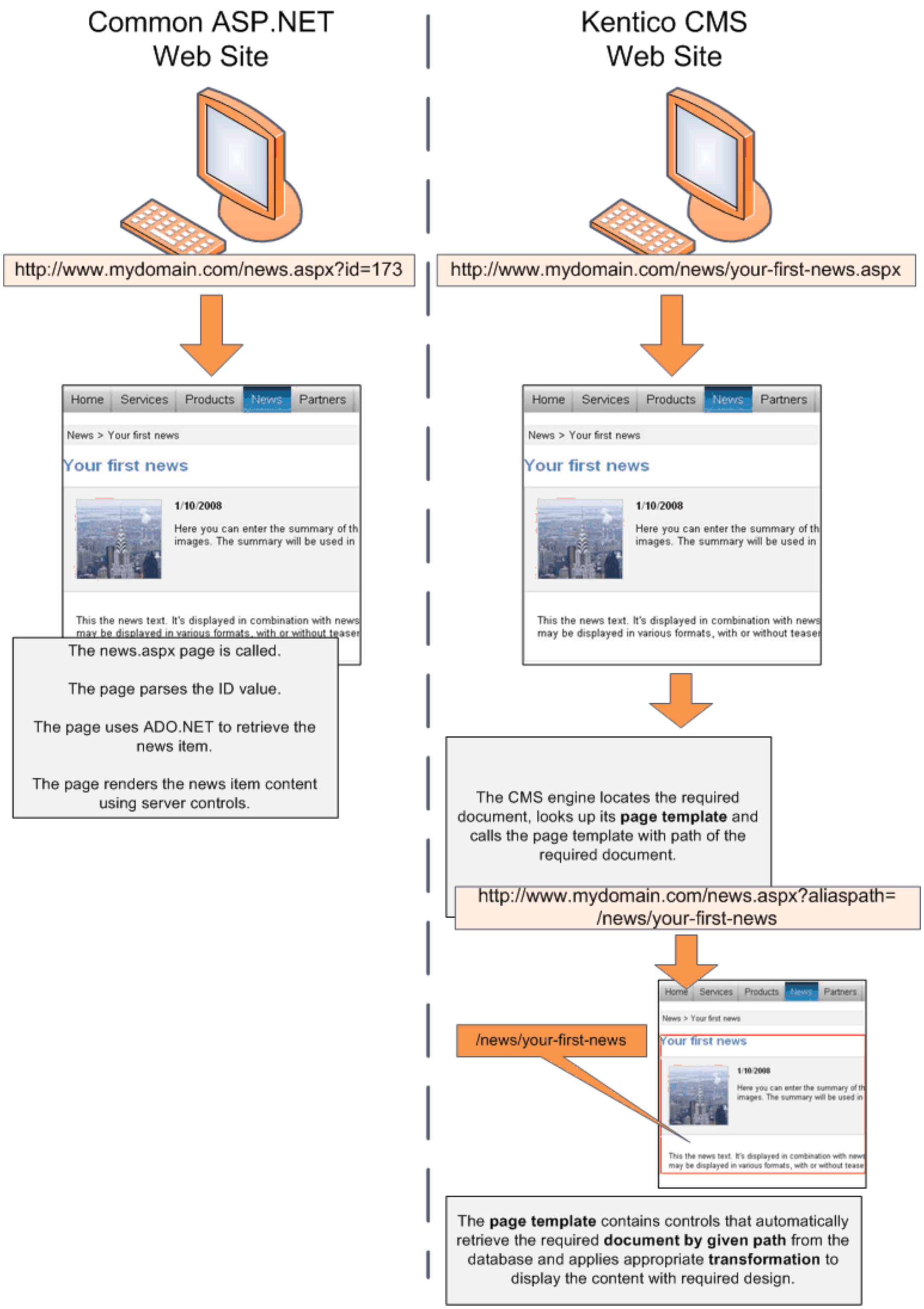
Please visit the Kentico CMS Showcase at http://www.kentico.com/Showcase.aspx for reference web sites, clients and testimonials.

## 2.2 How does it really work?

If you're familiar with dynamic web sites, you may want to know what the difference between a CMS system and a common ASP.NET web site 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:


Common ASP.NET
Web Site
Kentico CMS
Web Site
http://www.mydomain.com/news.aspx?id=173
http://www.mydomain.com/news/your-first-news.aspx
Home Services Products News Partners
News > Your first news
Your first news
1/10/2008
Here you can enter the summary of th
images. The summary will be used in
This the news text. It's displayed in combination with news
may be displayed in various formats, with or without teaser
The news.aspx page is called.
The page parses the ID value.
The page uses ADO.NET to retrieve the news item.
The page renders the news item content using server controls.
The CMS engine locates the required document, looks up its **page template** and calls the page template with path of the required document.
http://www.mydomain.com/news.aspx?aliaspath=
/news/your-first-news
/news/your-first-news
The **page template** contains controls that automatically retrieve the required **document by given path** from the database and applies appropriate **transformation** to display the content with required design.

### 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.***

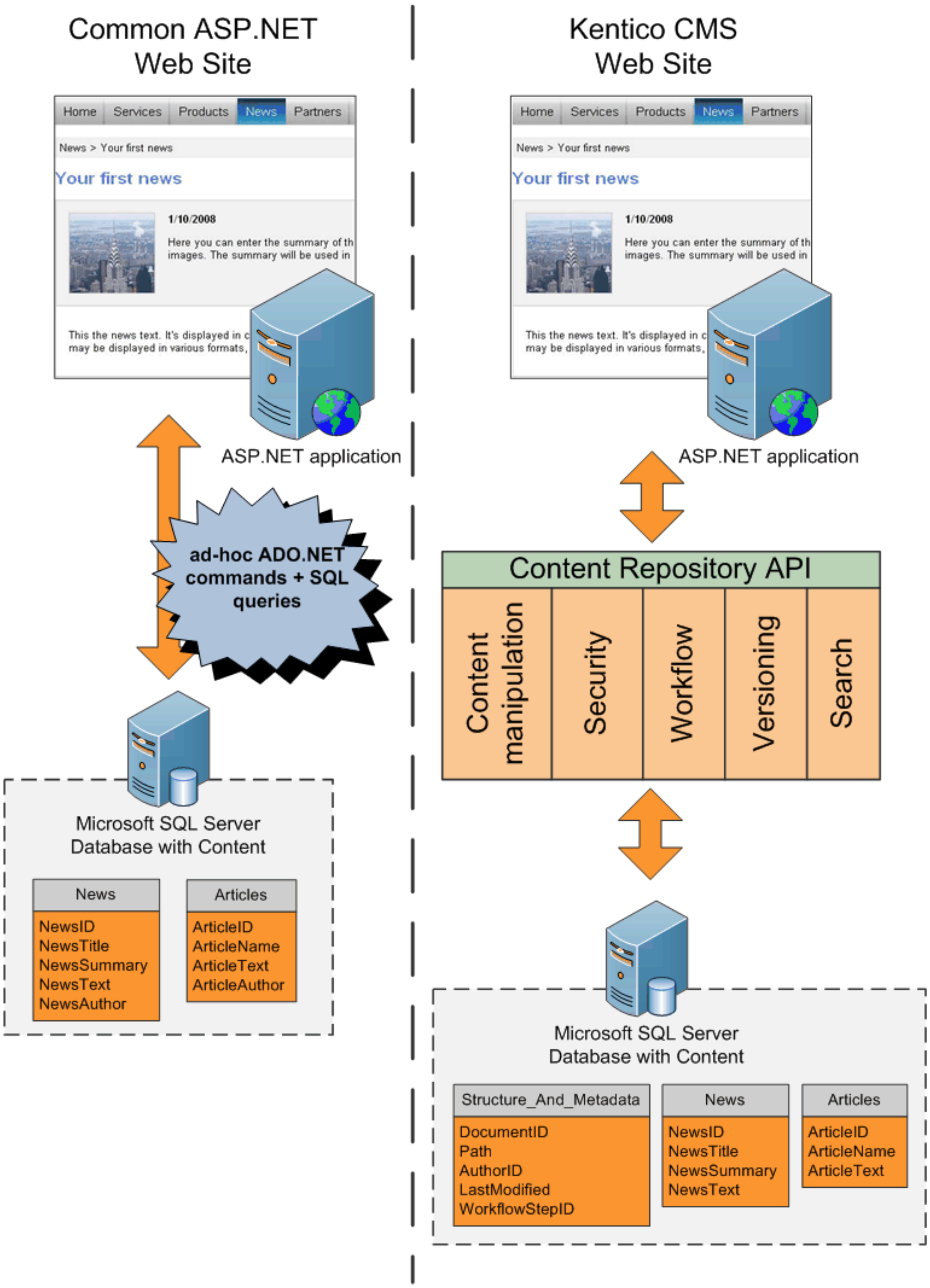
### What else does the CMS do for me?

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 web site 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 web sites 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

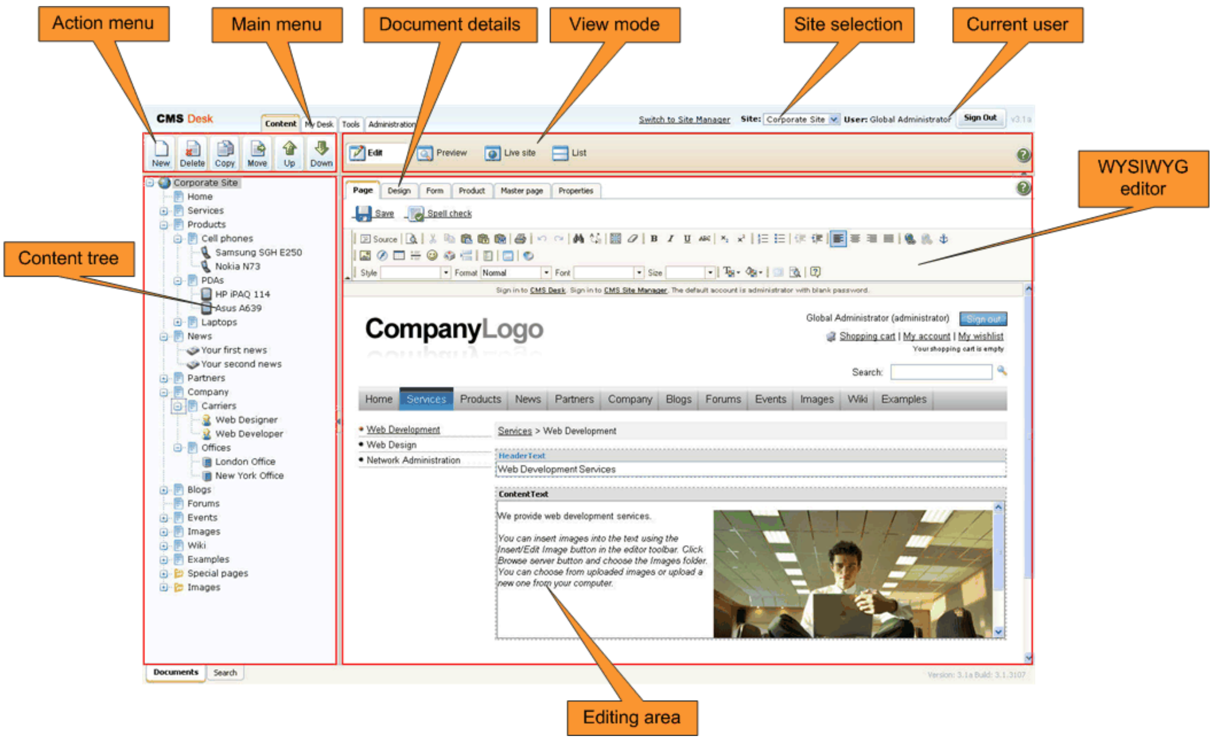
As you can see, the **content repository represents a systematic approach to content storage, manipulation and security**.

## 2.4 How do I edit content?

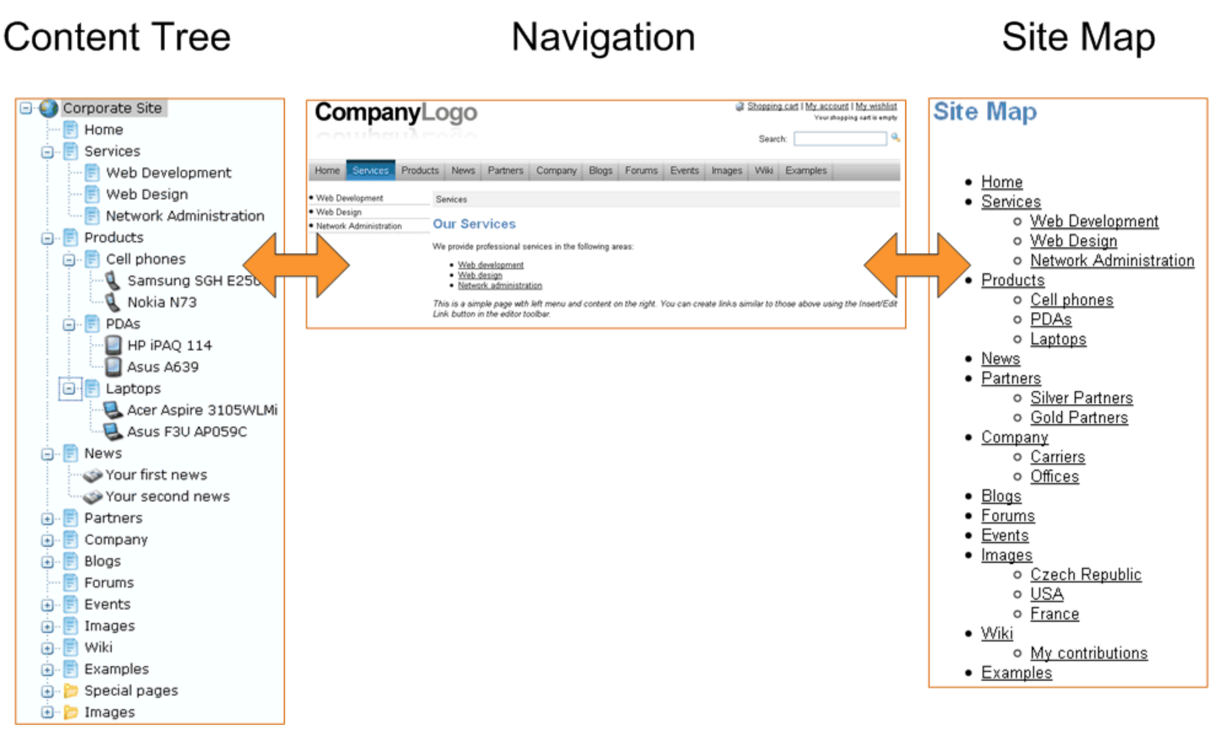
Until know, we were talking about the architecture and how the system works. But how do I edit the content? After all, it's the most important feature of a content management system.

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

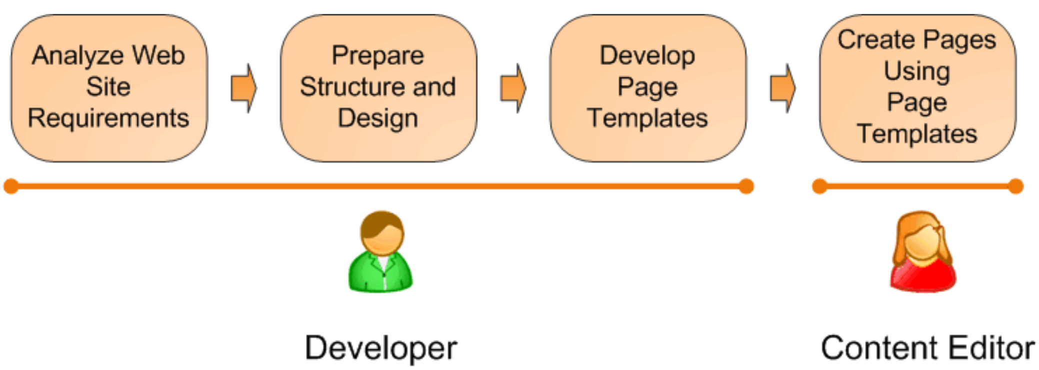
- **Edit** - editing mode
- **Preview** - preview mode. It displays the current version of the page before it's published (if you're using workflow). It also displays the content without using caching, which allows you to preview the content even if the live web site displays the cached version.
- **Live 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 web site?

Now that you know how to edit the content, you may want to know how to develop the web site 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 web site and how the roles are split between developers and editors. A typical development process consists of the following steps:

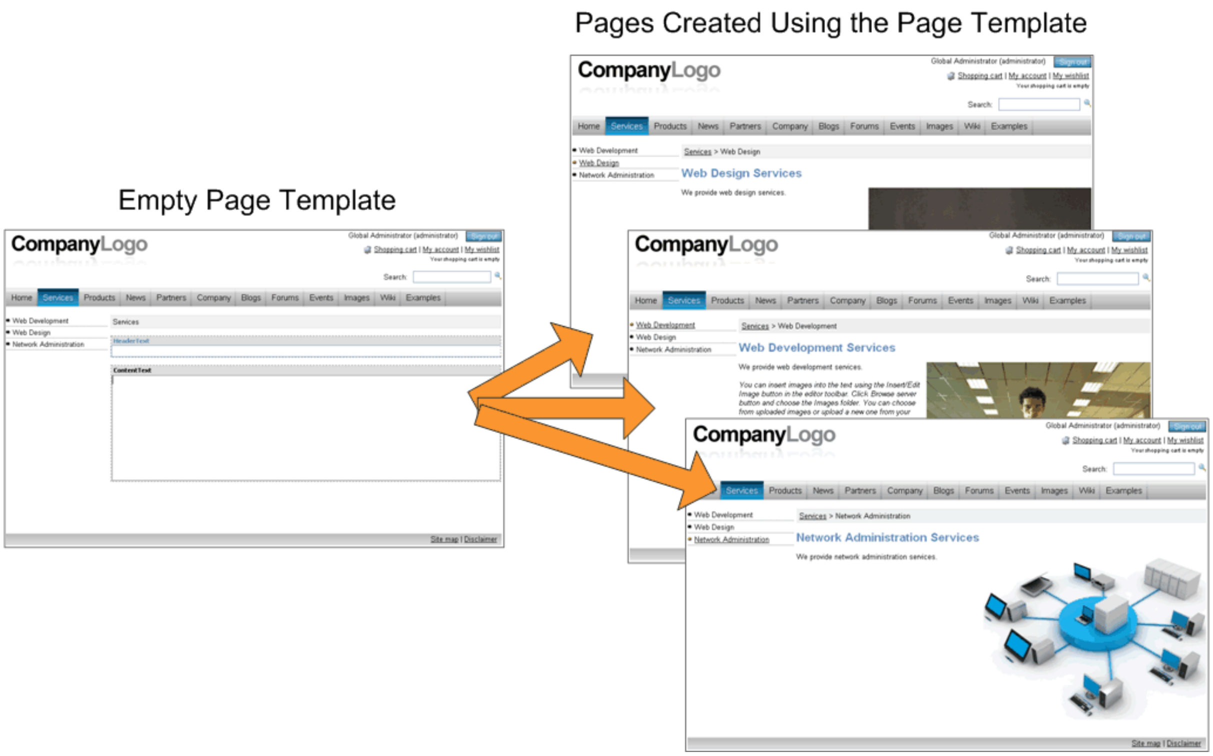
1. The developer analyzes the client requirements.
2. The developer prepares the site structure (site map) and web design.
3. The developer creates **page templates** for every type of the page (home page, solutions, products, news, etc.)

4. The content editor creates new pages - she enters text and images into the page templates defined by 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 web site. 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

# 3 Installation and deployment

## 3.1 Installation 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 web site 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 web 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 web site, you will need to create a virtual directory - see 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 chapter Troubleshooting installation issues.

**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 2005 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 with them.

**Web Server Requirements**

- One of the following operating systems:
  - Microsoft Windows 2000 Professional with MDAC 2.8 or later
  - Microsoft Windows 2000 Server with MDAC 2.8 or later
  - Microsoft Windows XP Professional with MDAC 2.8 or later
  - Microsoft Windows Server 2003 with MDAC 2.8 or later
  - Microsoft Windows Vista Business, Enterprise or Ultimate with MDAC 2.8 or later
  - Microsoft Windows Server 2008  with MDAC 2.8 or later
- .NET Framework 2.0 or later
- Microsoft Internet Information Services 5.0, 6.0 or 7.0 and ASP.NET 2.0 or later installed and correctly registered in IIS.
- Graphic adapter with resolution at least 1024x768 pixels.
- 100 MB free disk space.

ajax + shared hosting requirements

**SQL Server Requirements**

- Microsoft SQL Server 2000, MSDE 2000, SQL Server 2005 or SQL Server Express 2005, Microsoft SQL Server 2008. The server must use case insensitive (CI) collation.

**Development Machine Requirements**

- One of the following operating systems:
  - Microsoft Windows 2000 Professional with MDAC 2.8 or later
  - Microsoft Windows 2000 Server with MDAC 2.8 or later
  - Microsoft Windows XP Professional with MDAC 2.8 or later
  - Microsoft Windows Server 2003 with MDAC 2.8 or later
  - Microsoft Windows Vista Business, Enterprise or Ultimate
  - Microsoft Windows Server 2008 with MDAC 2.8 or later
- .NET Framework 2.0 or later
- Microsoft Internet Information Services 5.0, 6.0 or 7.0 and ASP.NET 2.0 or later installed and correctly registered in IIS.
- Graphic adapter with resolution at least 1024x768 pixels.
- 100 MB free disk space.
- Recommended:
  - Microsoft Visual Studio .NET 2005/2008 or Microsoft Web Developer 2005/2008.
  - 2 GB RAM
  - Pentium IV or Pentium Core 2 Duo

**Web Client Requirements**

- **The Web site** created using Kentico CMS can be viewed on virtually any browser depending on the code you generate. Some of the Kentico CMS Controls (such as DHTML drop-down menu) are limited to higher-level browsers only as described in the Kentico CMS Web Parts and Controls Reference.
- **Kentico CMS Desk** (the administrative user interface for administrators, developers and site contributors) was tested with the following browsers running on Microsoft Windows XP SP2:
  - Internet Explorer 6.0 or later
  - Mozilla 1.7.1 or later
- If you're using **Mac OS** as a web client, you can use Safari 3.0 or Firefox 2.0 to access the web site and administration interface.
- The administration interface requires a graphic adapter with resolution at least 1024x768

pixels.

**Shared Web Hosting Requirements**

- Support for ASP.NET 2.0 or later.
- Microsoft SQL Server 2000, MSDE 2000, SQL Server 2005 or SQL Server Express 2005, Microsoft SQL Server 2008.
- The server must use case insensitive (CI) collation and the IDENTIFIER_CASE parameter must be set to MIXED.
- The server must support medium-trust or full-trust security level.
- If the server uses medium trust, ASP.NET AJAX 1.0 must be installed on the server.

## 3.3 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 the chapter Installation on shared hosting server to find how to solve this.

Run KenticoCMS.exe. You will see the welcome screen:


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

Choose the location where Kentico CMS web installer and documentation will be deployed. Please note: this is not the folder where your web site will be placed, it's only a place for Kentico CMS program files and help files. Click **Next** and then **Install**.

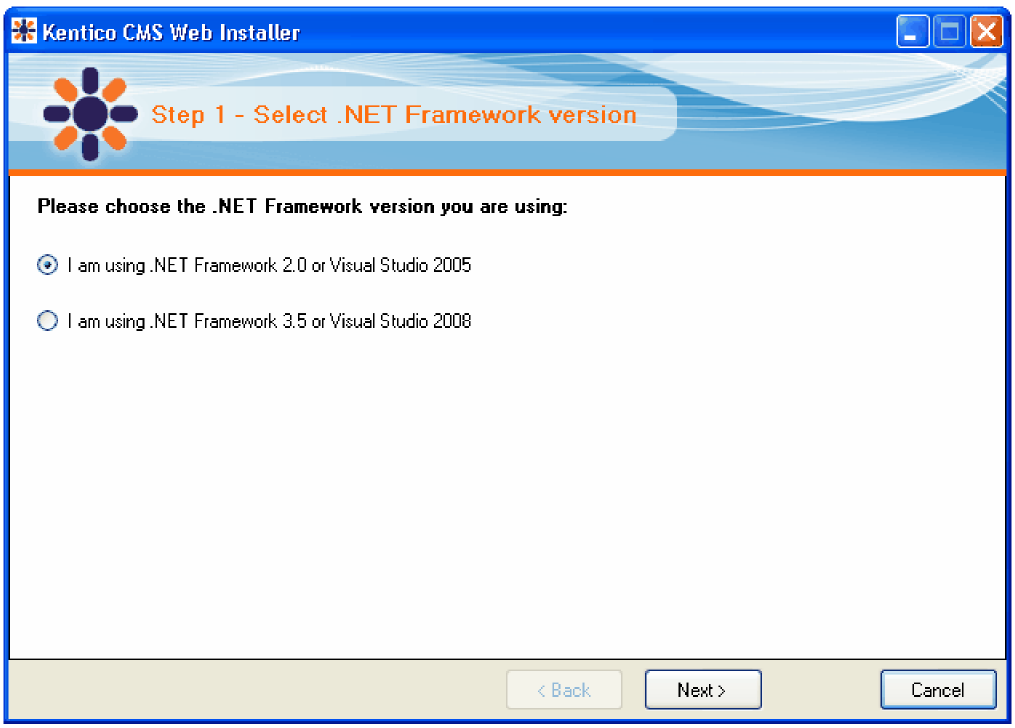
After the installation is finished, choose to launch Kentico CMS Web Installer and click **Finish**. Continue to Kentico CMS Web Installer.

## 3.4 Web Installer

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

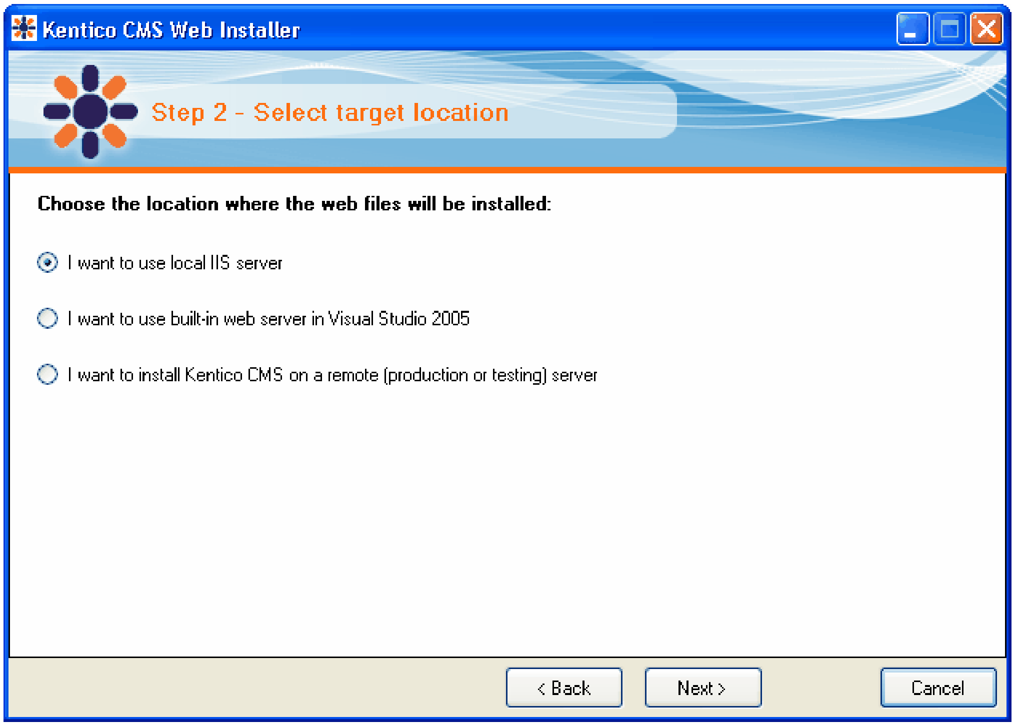
**Step 1 - Select .NET Framework version**

First, you need to choose whether you use Visual Studio 2005 or Visual Studio 2008. Depending on your choice, the installer will use appropriate web.config and webproject.sln files. The binaries and code are same for both options; they are compiled for .NET 2.0 and can be used with .NET 3.5 as well.



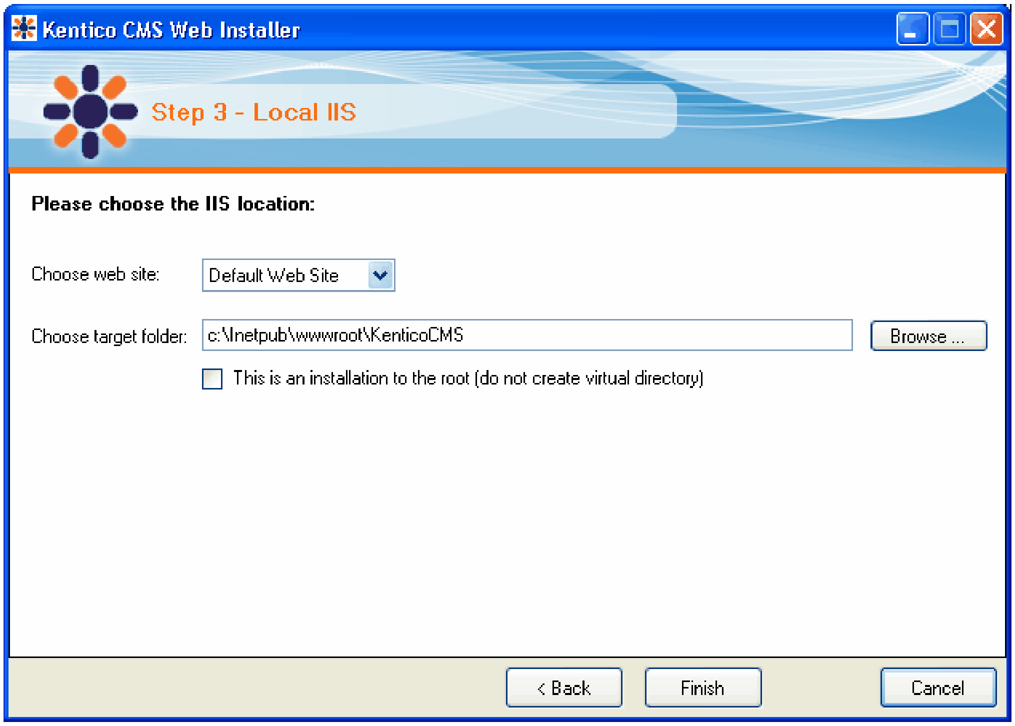
### Step 2 - Choosing the target location

Choose one of the options:
**- installation on IIS server** - you must have local IIS server installed, configured for ASP.NET 2.0 and running
**- Visual Studio 2005 built-in web server** - you must have Visual Studio 2005 or Visual Web Developer 2005 Express Edition installed on your local machine
- **remote 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)



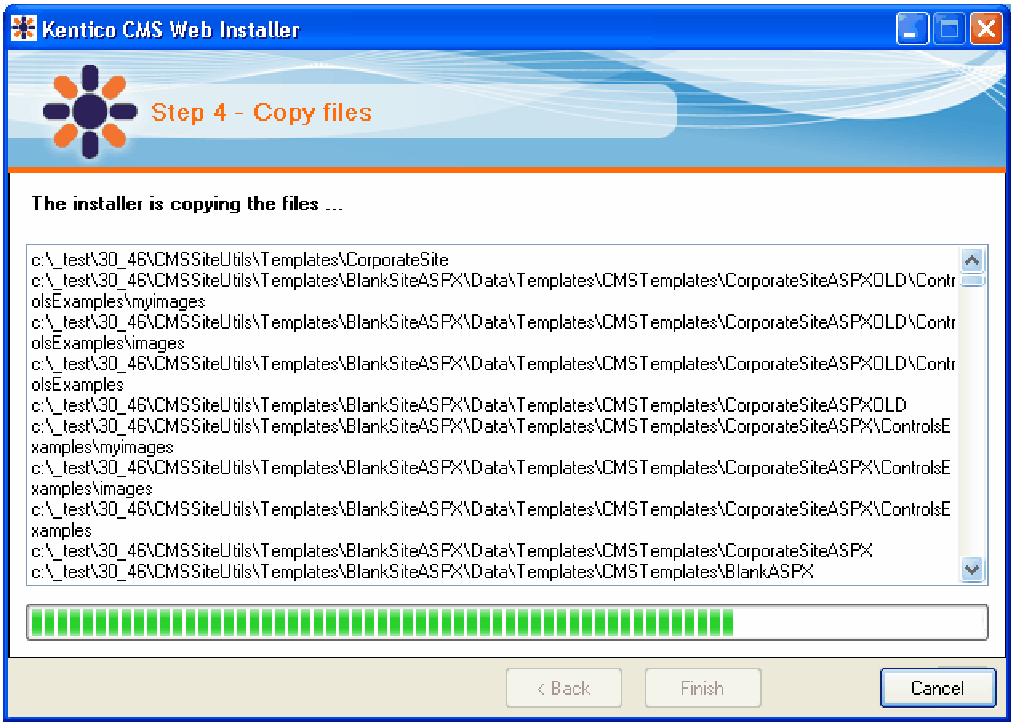
## Option 1 - Installation on your local IIS server

If you run local IIS server on your machine, choose the first option in Step 2 and click **Next**. In step 3, you need to choose one of the web sites configured on your IIS and choose disk folder where files will be placed. Please make sure that the web site you choose is running.

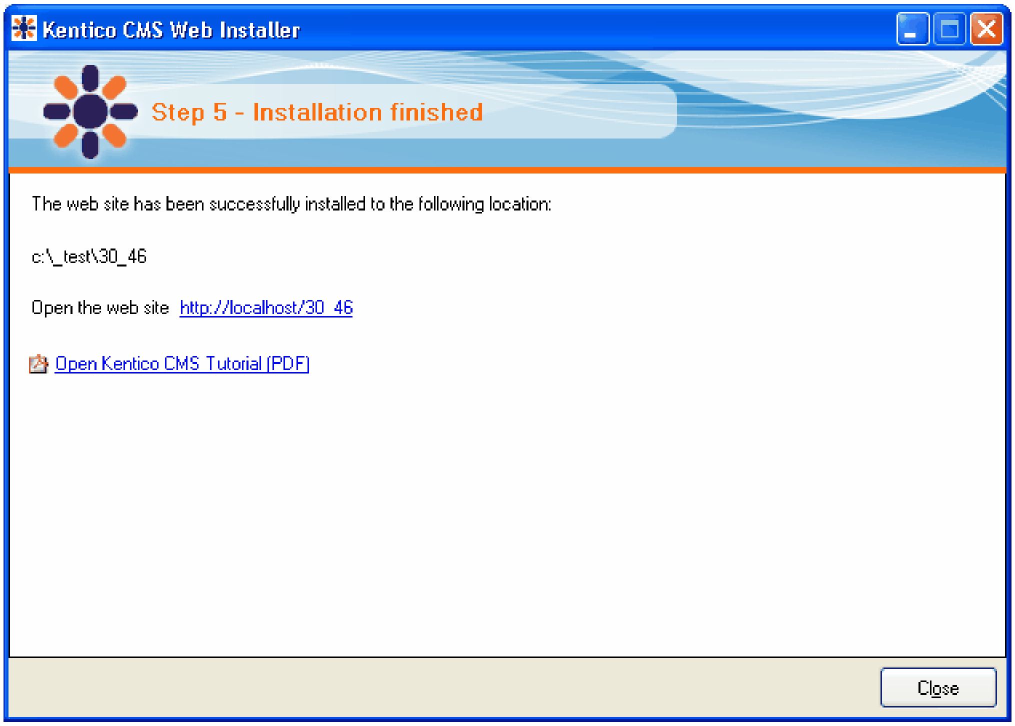



**Please note:** if you're installing Kentico CMS into the root of your web site (such as http://www.domain.com) and do not wish to create a virtual directory (such as http://www.domain.com/cms), please check the box **This is an installation to the root**.

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

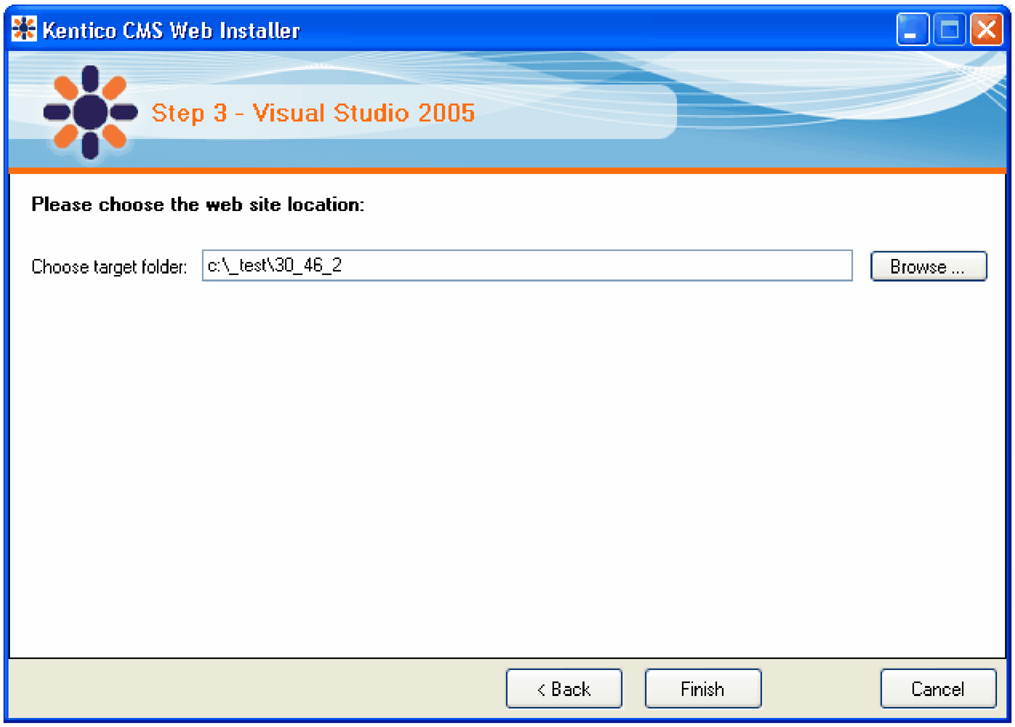


At the end, you're displayed with URL of the new Kentico CMS installation. Please put down this URL for future use.

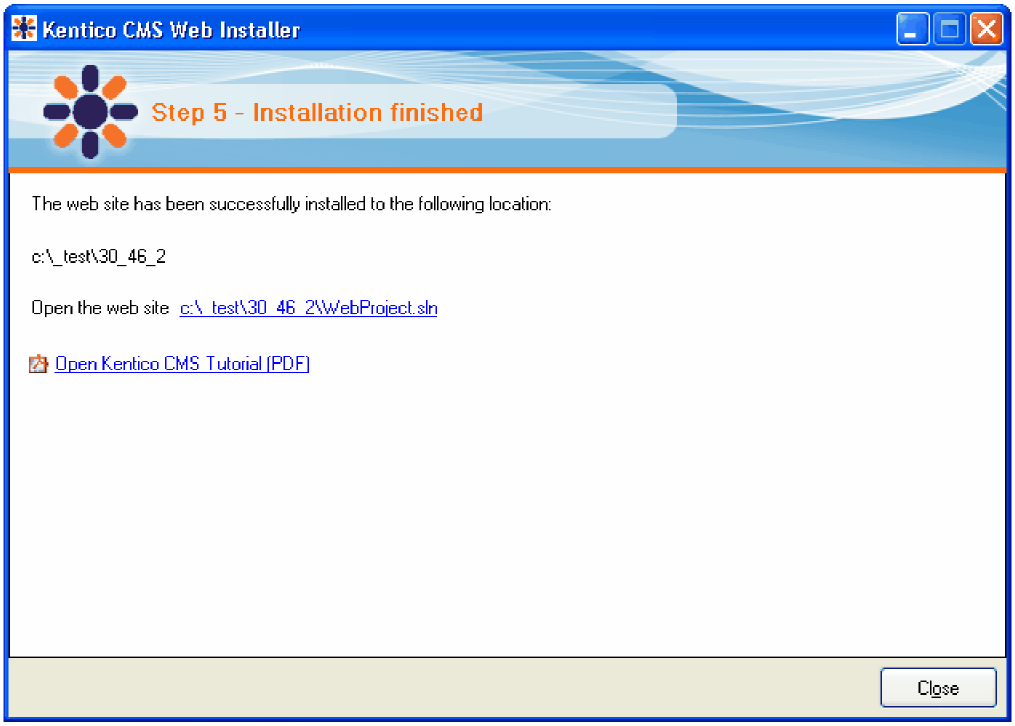


## Option 2 - using the built-in Visual Studio 2005 web server

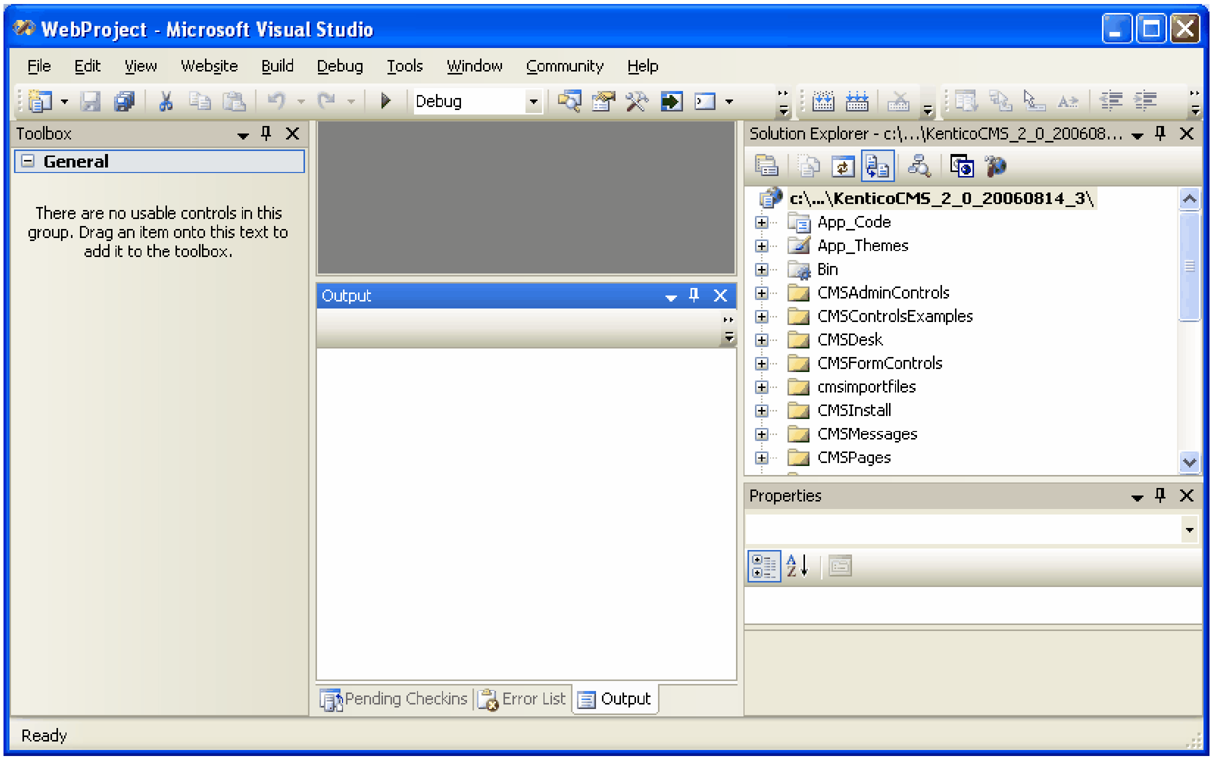
If you want to use only the built-in web sever in Visual Studio 2005 (or Visual Web Developer 2005), choose the appropriate option and click Next. You will need to enter the target location of the files on your disk:



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



The project will open in Visual Studio:



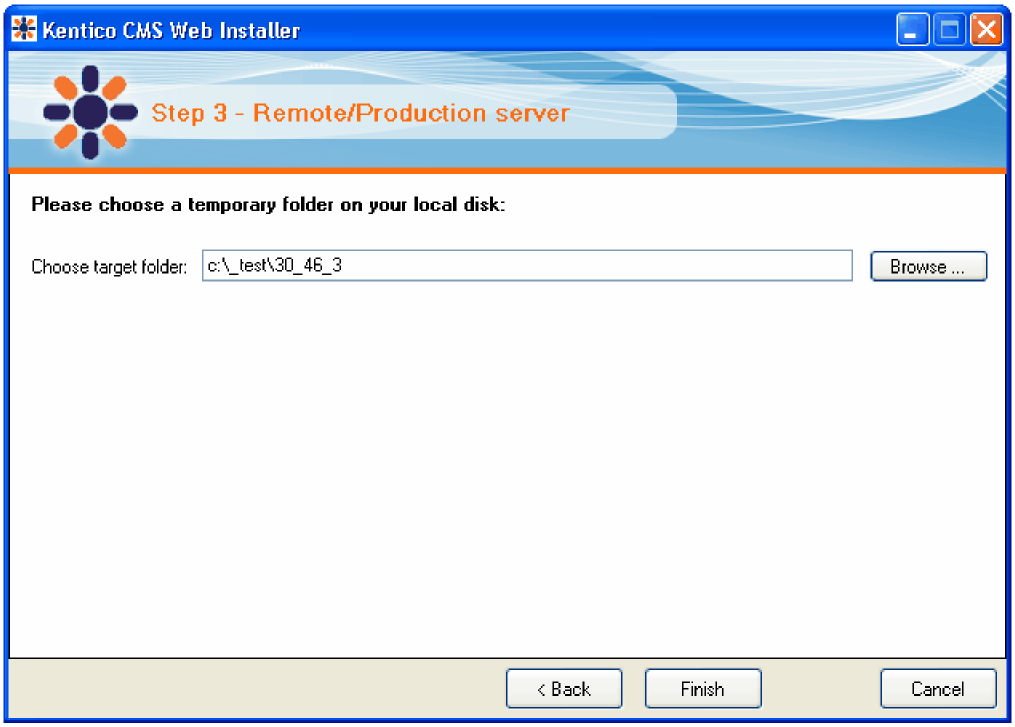

Choose **Debug -> Start without debugging** from the main menu. The site is displayed in the new browser window, using the built-in VS 2005 web server.

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

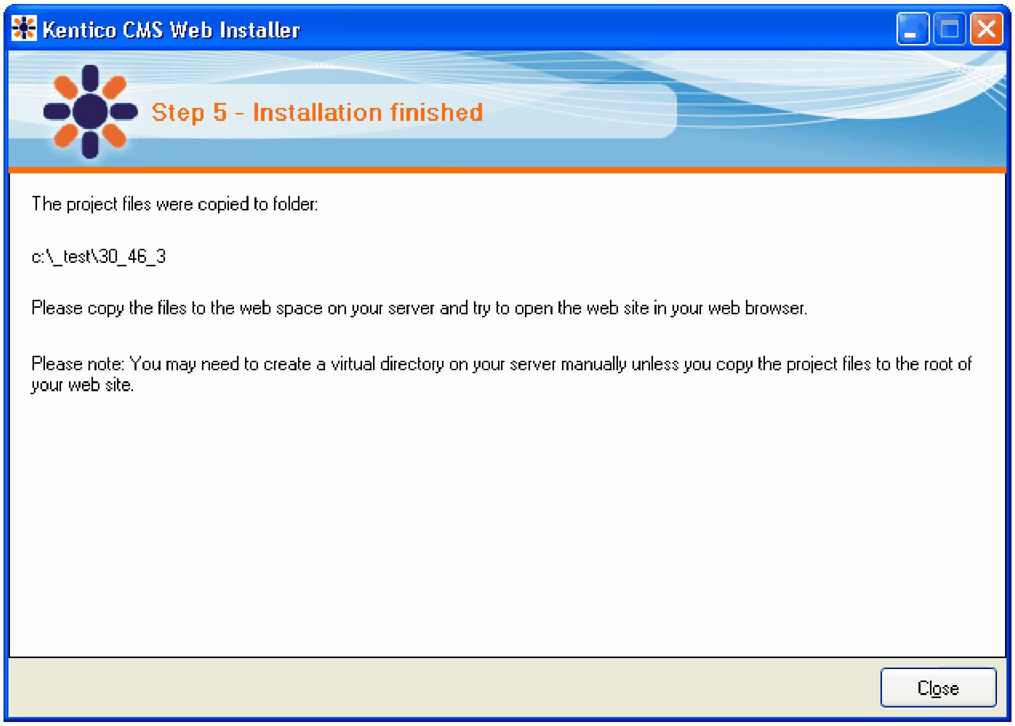
If the link for opening the project in Visual Studio doesn't work, you may need to start Visual Studio manually and choose **File -> Open -> Web Site...** and locate the project folder on your disk manually.

## Option 3 - Installation on the 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 can choose the last option - installing Kentico CMS on a remote (production) server. You will need to specify a temporary folder on your local disk where the web project will be created:



Click Finish. Now the web project is created on your disk and a confirmation message is displayed:




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

Now you can continue to Database setup wizard.

## 3.5 Database setup

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

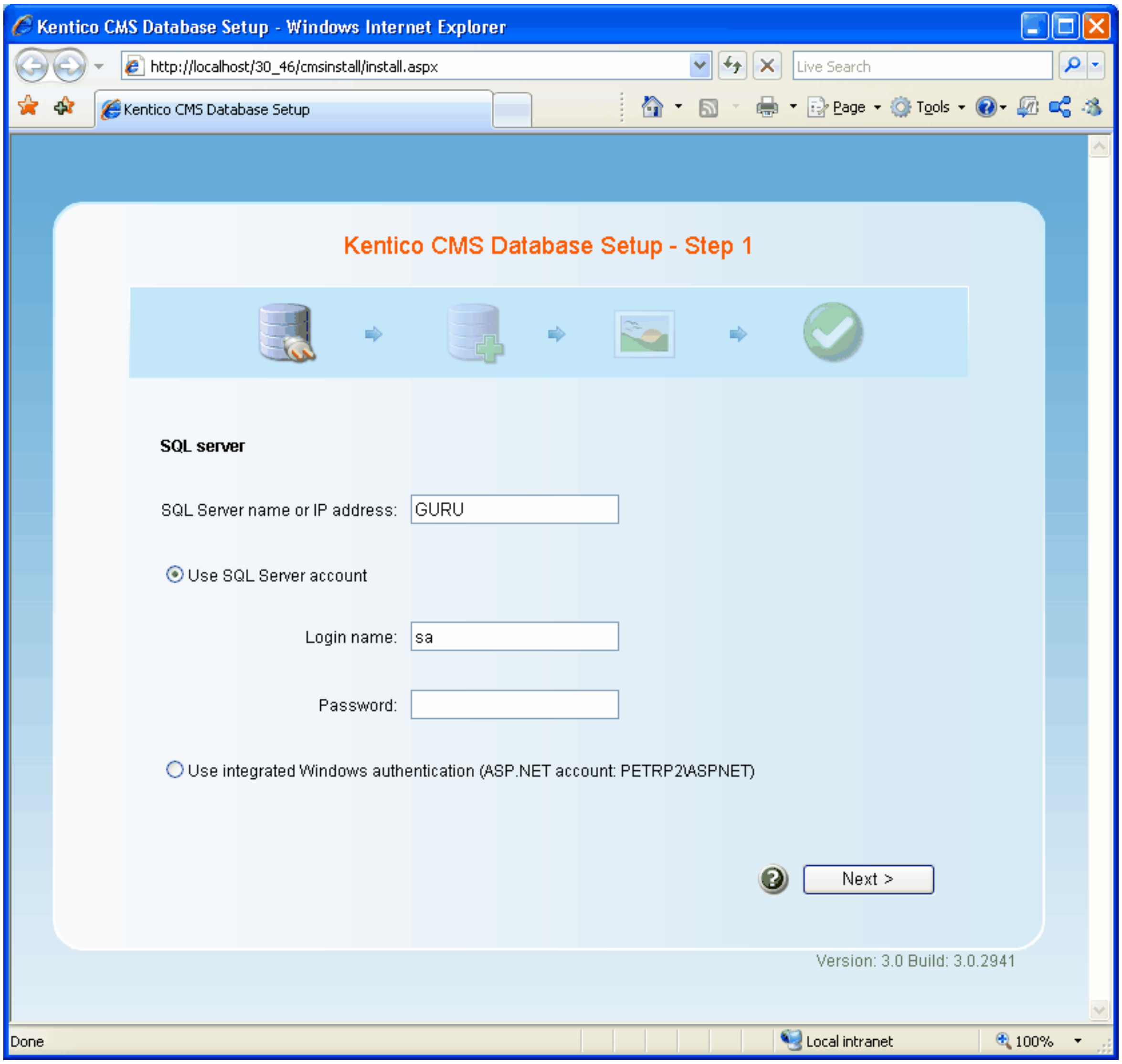
In Step 1, you need to specify the Microsoft SQL Server and login:

**- 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

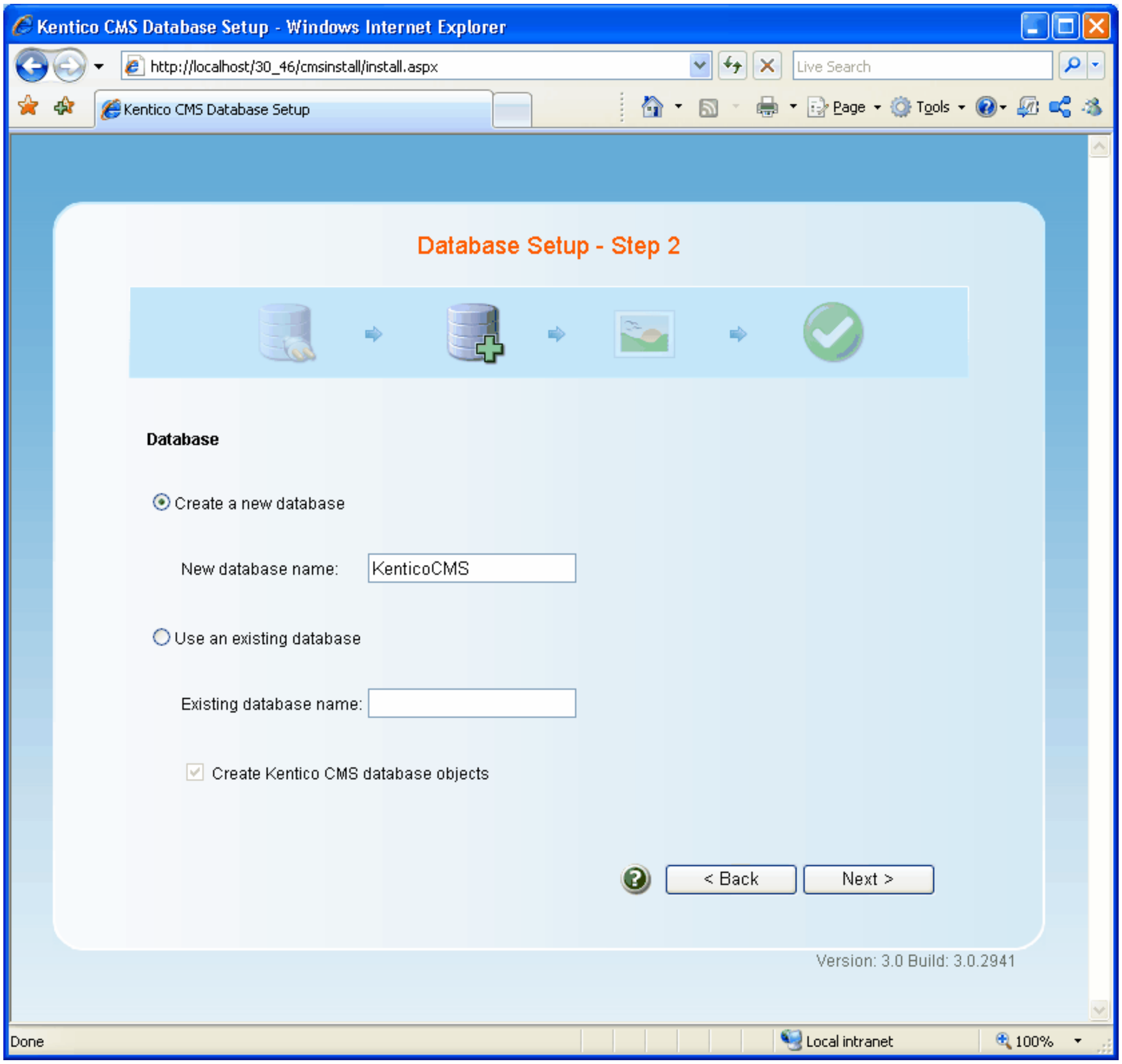
**- Use integrated Windows authentication (ASP.NET account)** - use this option if your server is configured for Windows integrated authentication. In this case, you need to use SQL Server 2005 Management Studio (or SQL Server 2000 Enterprise Manager) to create a new login for user account under which you currently run the web application (ASPNET for Windows XP and Network Service for Windows 2000/2003 - the actual ASP.NET account name is displayed on the screen).

Click the **Next** button.

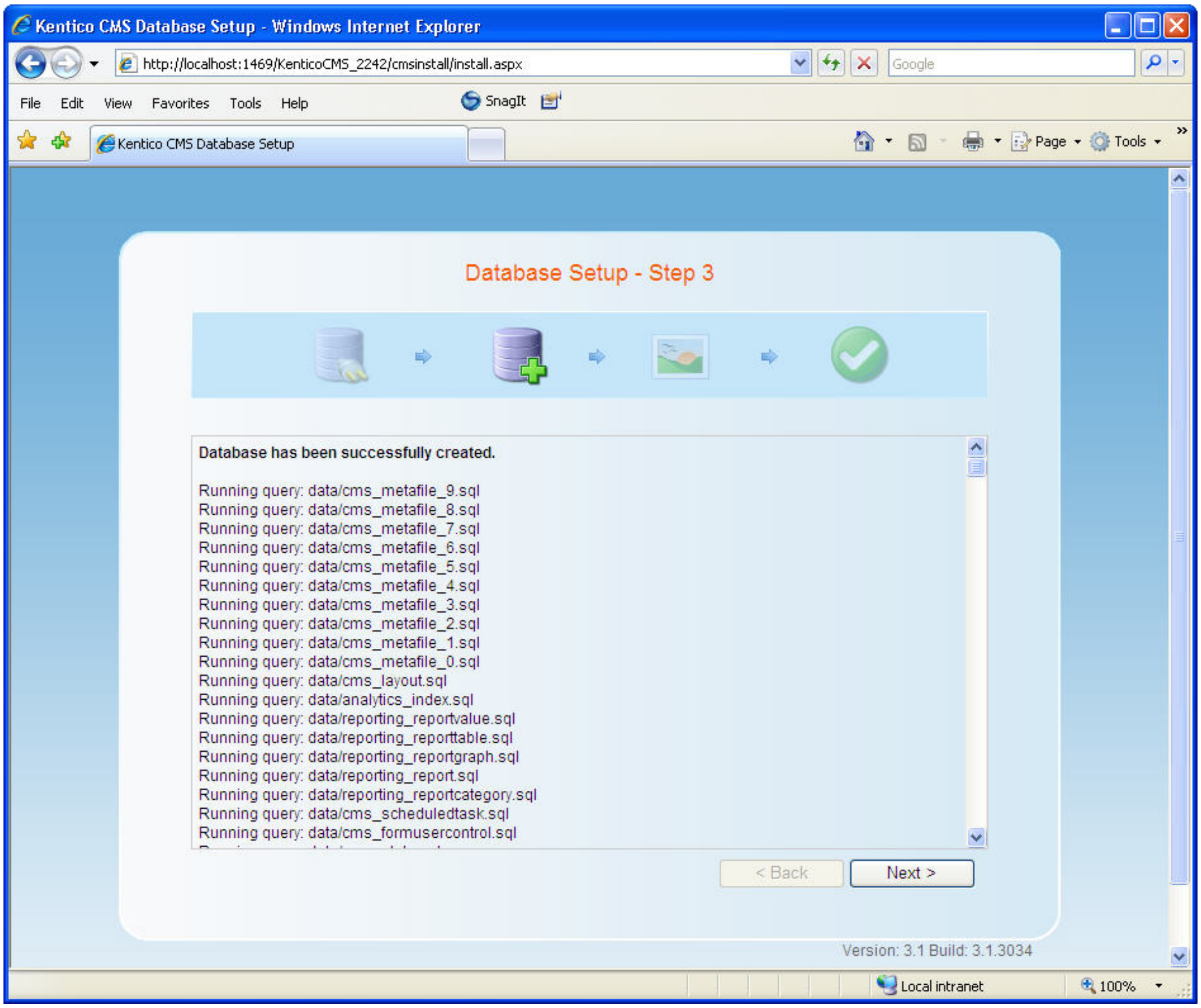

Kentico CMS Database Setup - Windows Internet Explorer
http://localhost/30_46/cmsinstall/install.aspx
Kentico CMS Database Setup
Kentico CMS Database Setup - Step 1
SQL server
SQL Server name or IP address: GURU
Use SQL Server account
Login name: sa
Password:
Use integrated Windows authentication (ASP.NET account: PETRP2\ASPNET)
Next >
Version: 3.0 Build: 3.0.2941
Done
Local intranet
100%

Now you can decide if you want to use an **existing database** or **create a new database**.

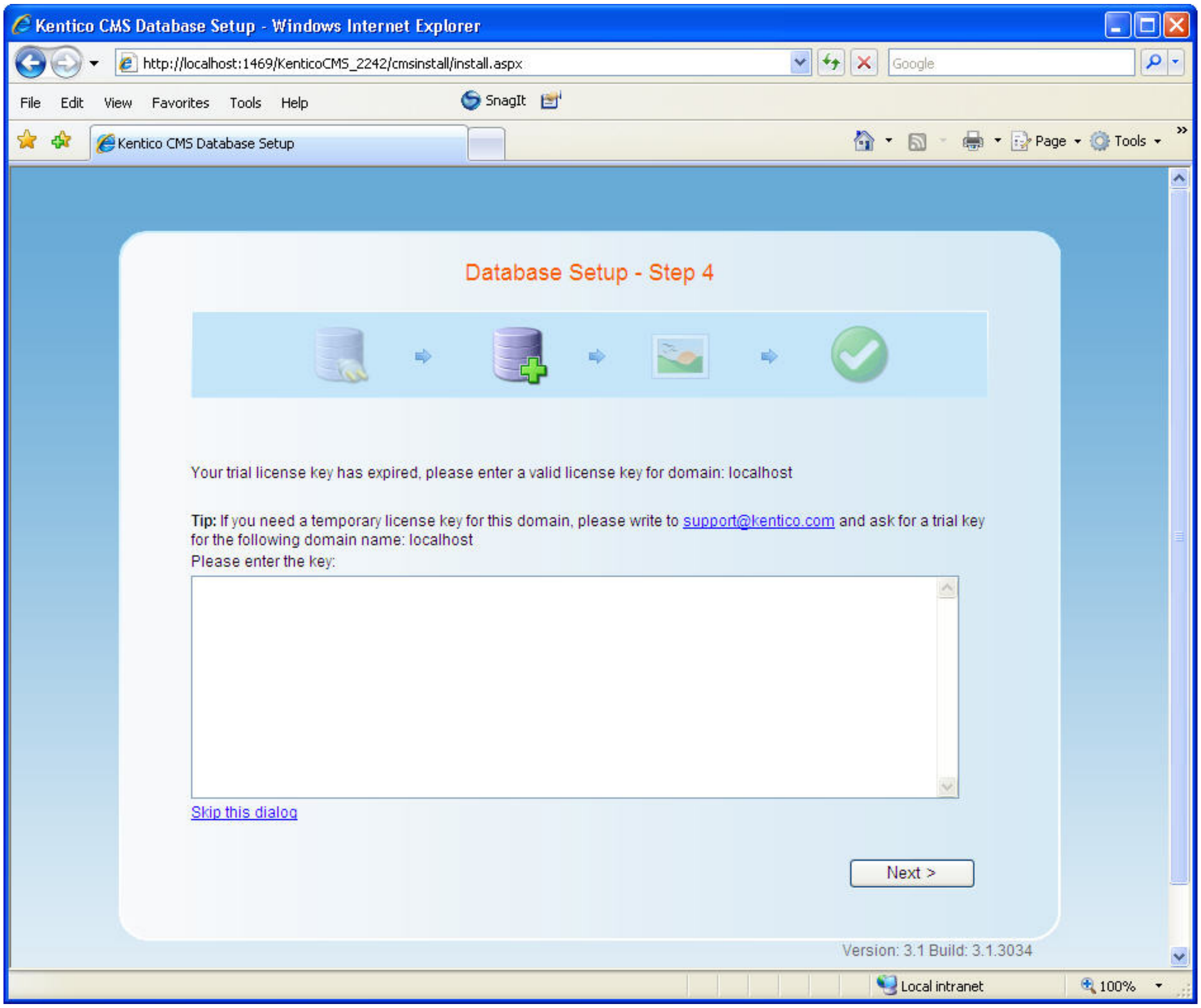
In case of using an existing database, you can choose to create the database tables, stored procedures and views - If the existing database already contains Kentico CMS objects, you need to uncheck the box **Create Kentico CMS database objects**. Click **Next.**



A log will be displayed, showing the progress of creating the database. 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.



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. Please note that you can skip this dialog and go to CMS Site Manager/Sites/New site wizard by clicking **Skip this dialog** at the bottom left.

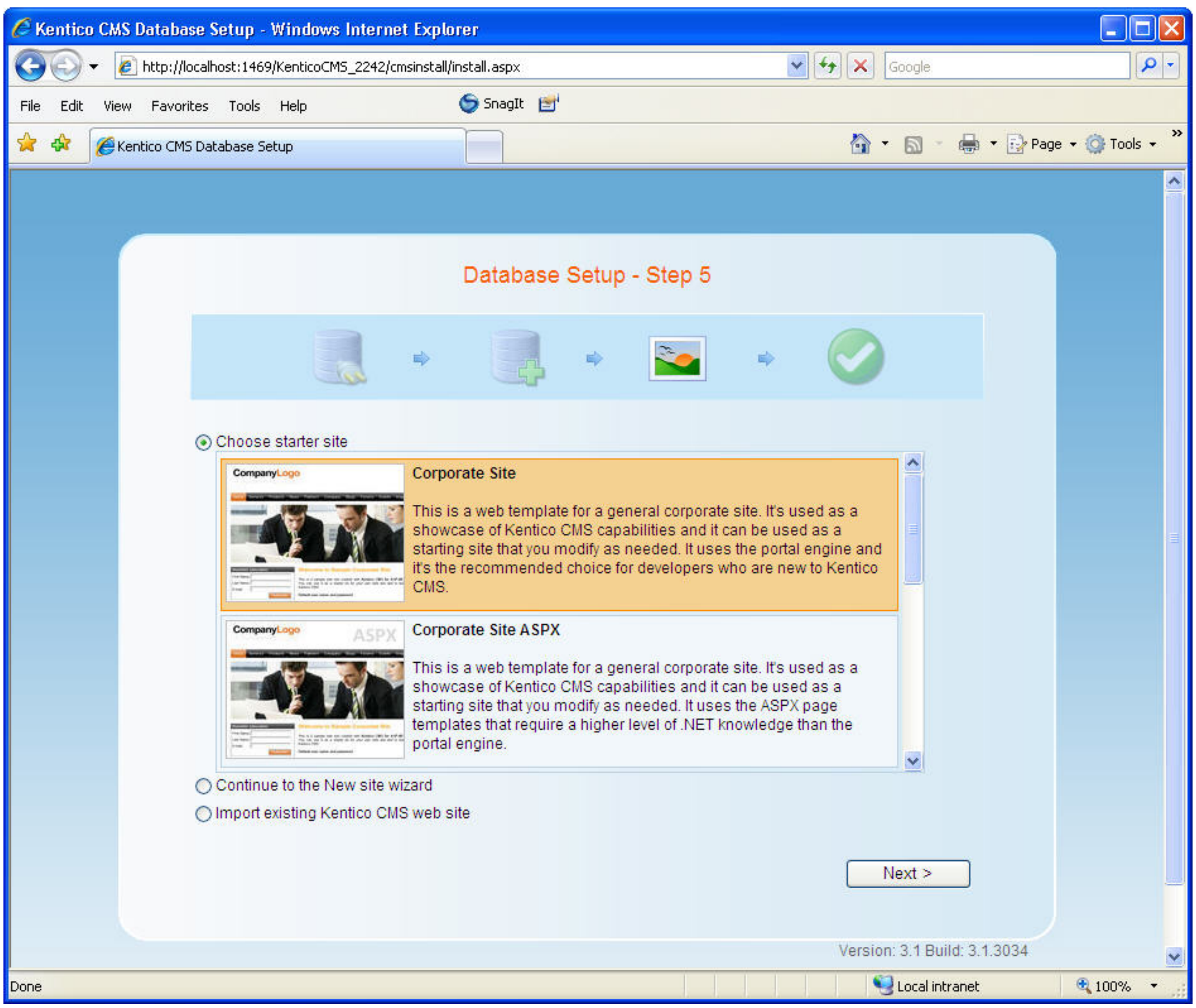


On the next screen, you will be offered with the following options:

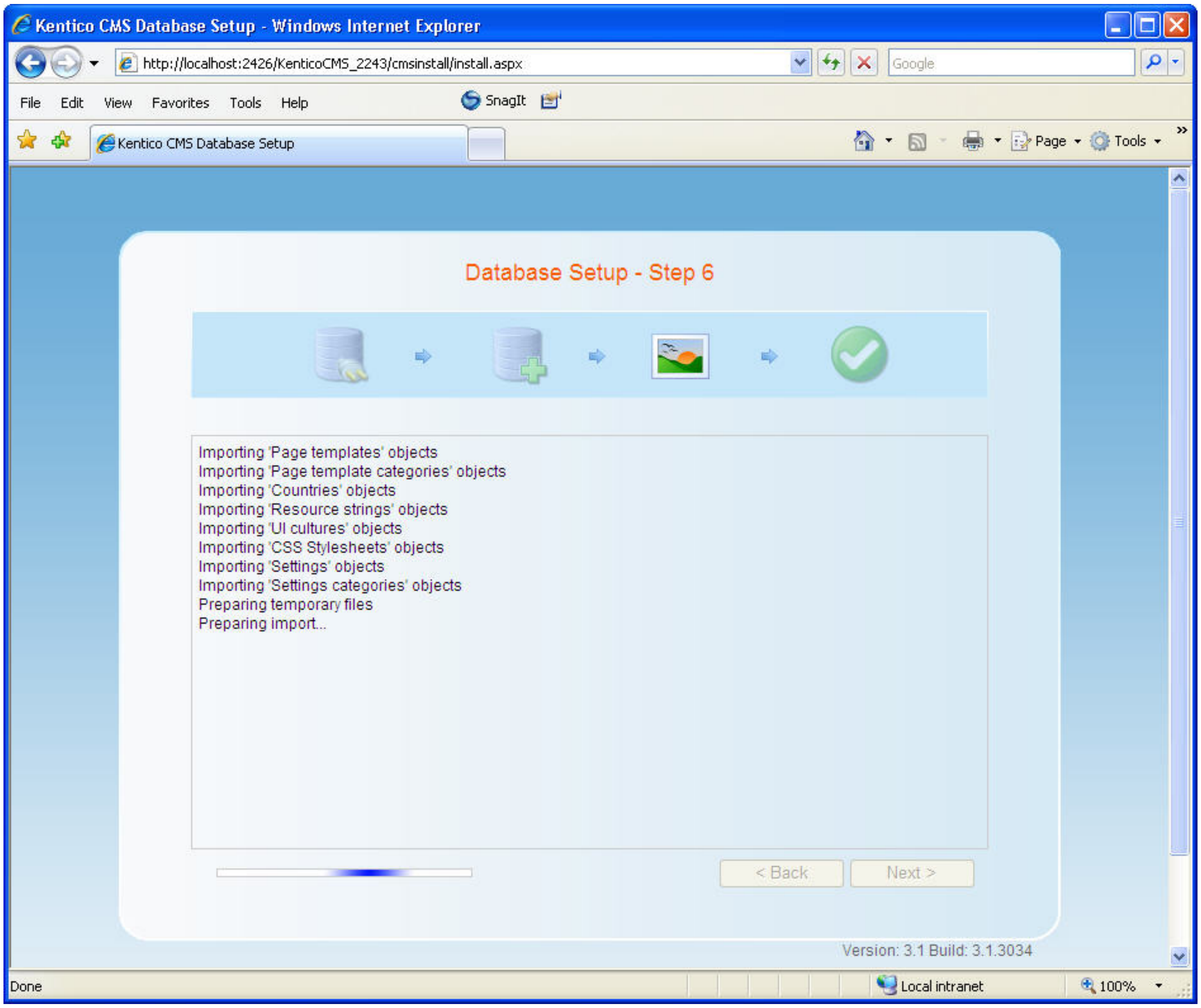
- **Choose a starter site such as:**
  **- Corporate Site** - this option installs the sample corporate site - it is recommended for most users, especially for evaluators.
  **- Corporate Site ASPX** - this option is recommended only for experienced ASP.NET developers who want to use ASPX page templates 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 other

- **Continue to the New site wizard** - this option is recommended if you're starting a new site from scratch.
- **Import an existing Kentico CMS web site** - use this option is you already created a web site with Kentico CMS and need to import it into a new installation (e.g. on the production server).

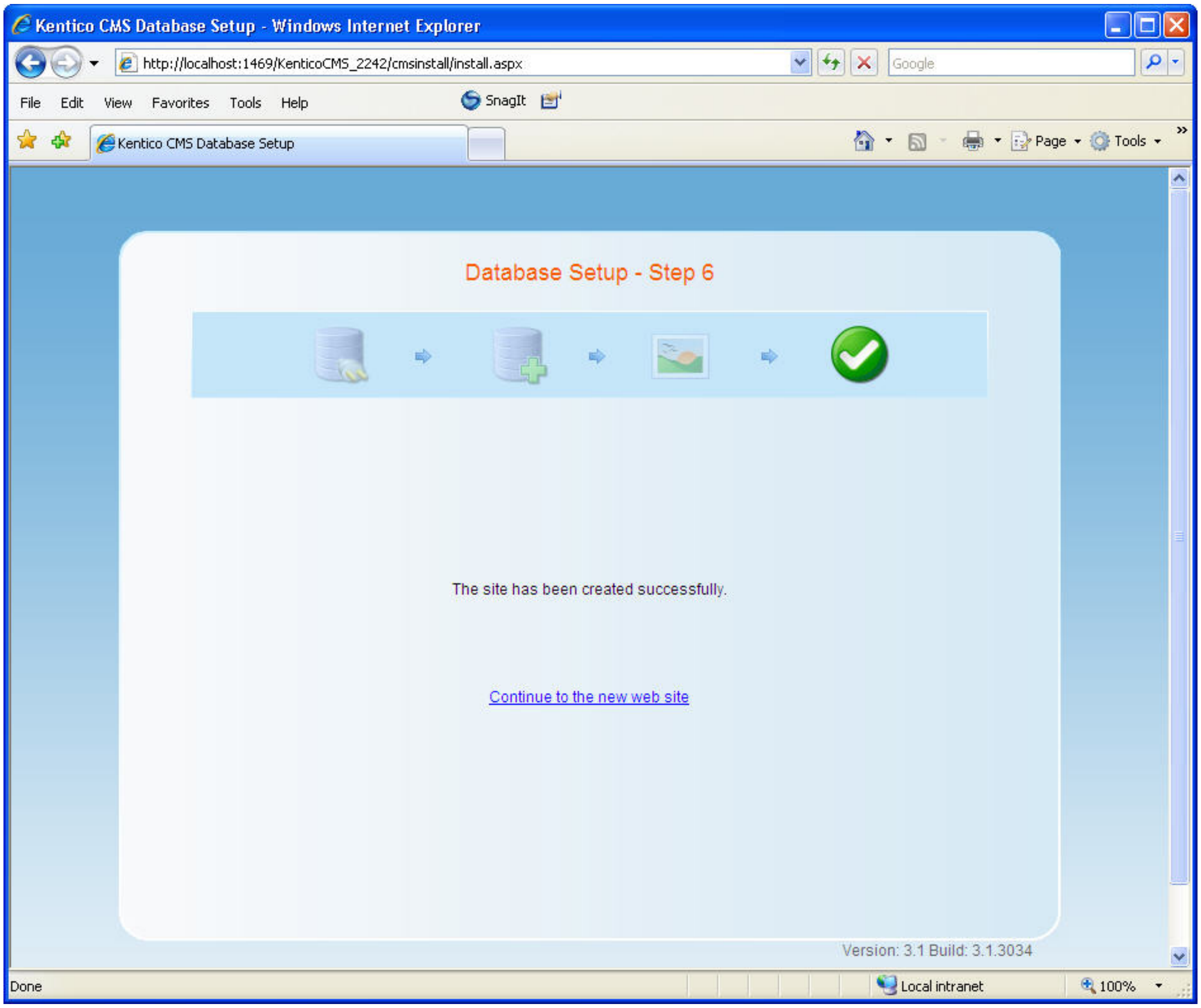
If you're new to Kentico CMS, it's highly recommended that you start with sample Corporate Site (portal engine). If you decide to run the New site wizard, you can find more details in the following chapter. Select an option and click the **Next** button.



A log will be displayed in the next step, showing you the progress of creating the web site.



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




**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.

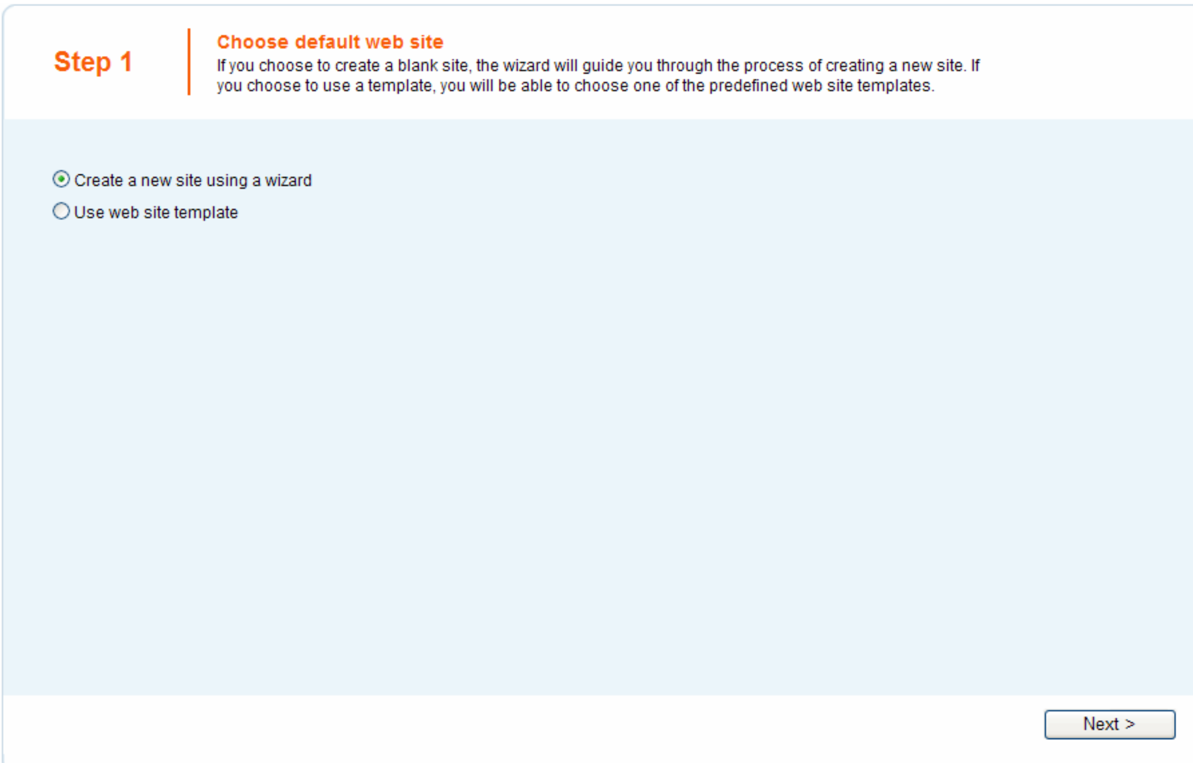
The default URL is http://localhost/KenticoCMS.
The default URL of CMS Desk is http://localhost/KenticoCMS/CMSDesk.
The default URL of Site Manager is http://localhost/KenticoCMS/CMSSiteManager.

## 3.6 New site wizard

The New site wizard guides you through the process of creating a new web site. You can start it in **Site manager -> Sites -> New site wizard**.

**Step 1**

In the first step of the wizard, you can select if you want to create the new site using a wizard or use a web site 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 web site template** - creates a new site based on a template chosen in the next step.

Select one of the options and click the **Next** button.

*Depending on the settings made in this step, the wizard can continue in two different ways.*

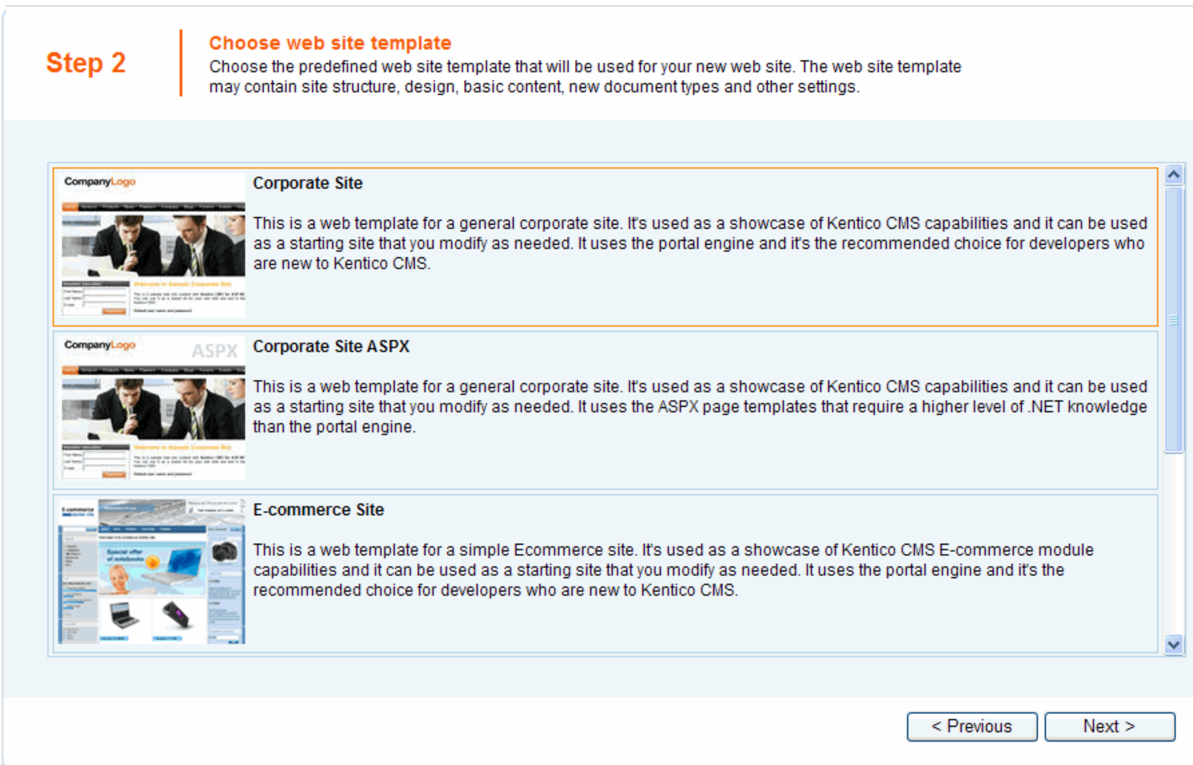
## 'Use web site template' version:

### Step 2

In Step 2, you can choose from a number of web site 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.
- **Blank site** is a blank template used for creating websites from scratch.
- and others.

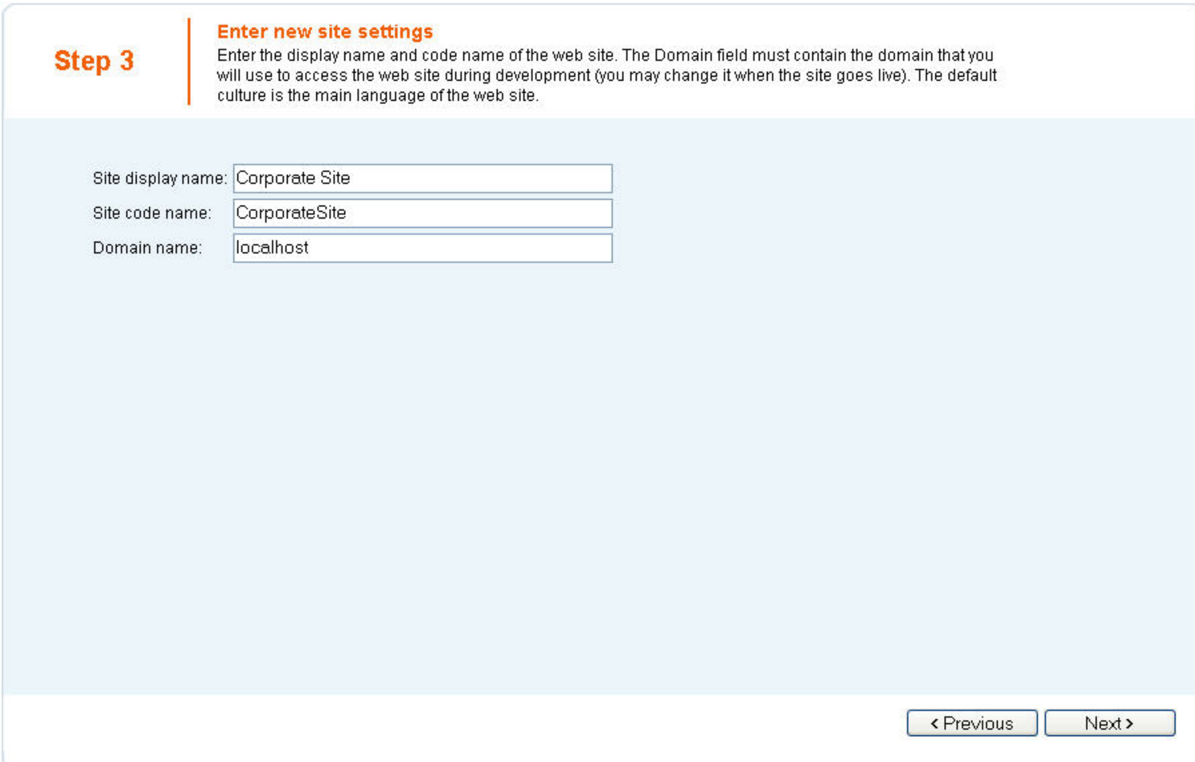
Some of the templates are available in two versions, one using the portal engine and the other using ASPX page templates.




Choose one of the offered sites and click the **Next** button.

**Step 3**

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.



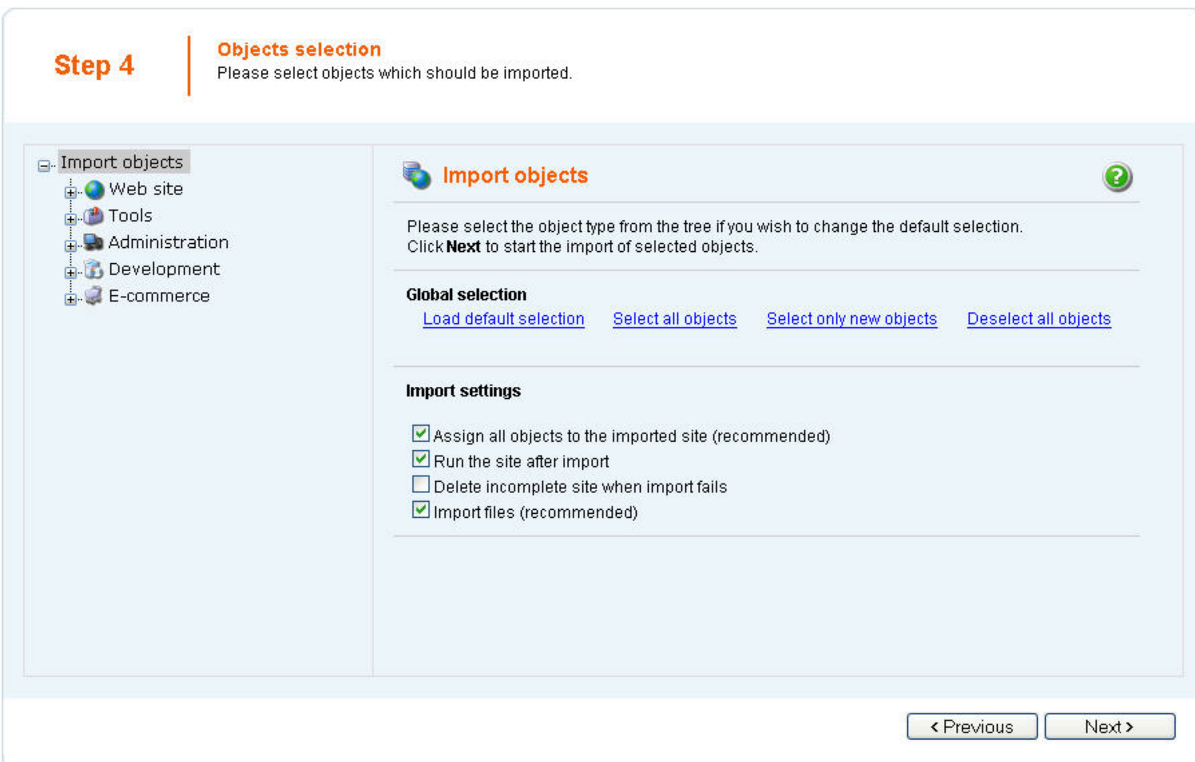

Click **Next** to continue.

**Step 4**

In Step 4, you can select which of the objects from the import package are to be imported and which are not. 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 which objects are to be imported and which are not.

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.

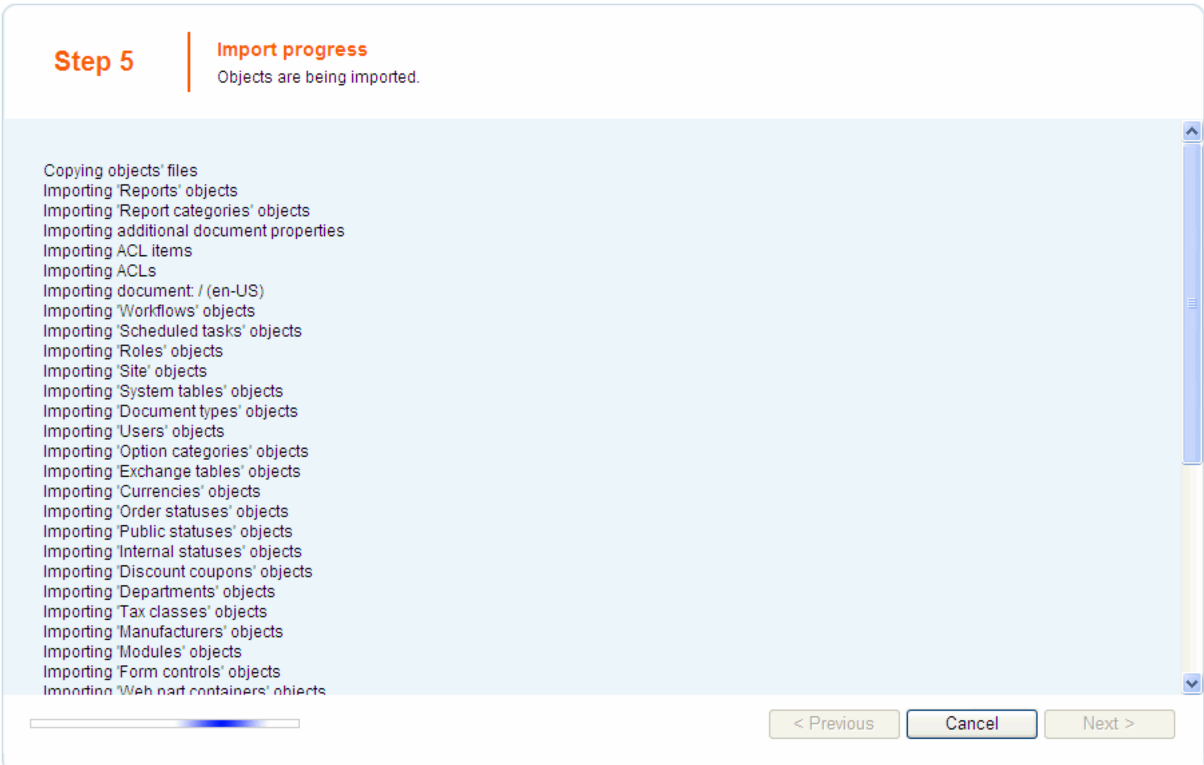
**Import settings**
- **Assign all objects to the imported site (recommended)** - if checked, all imported site related objects will be assigned to the imported site.
- **Run the site after import** - if checked, the updated site will be run after the import is finished.
- **Delete incomplete site when import fails -** if checked, incomplete site will be deleted when import fails.
- **Import files (recommended)** - if checked, files will be imported.




Click **Next** to continue.

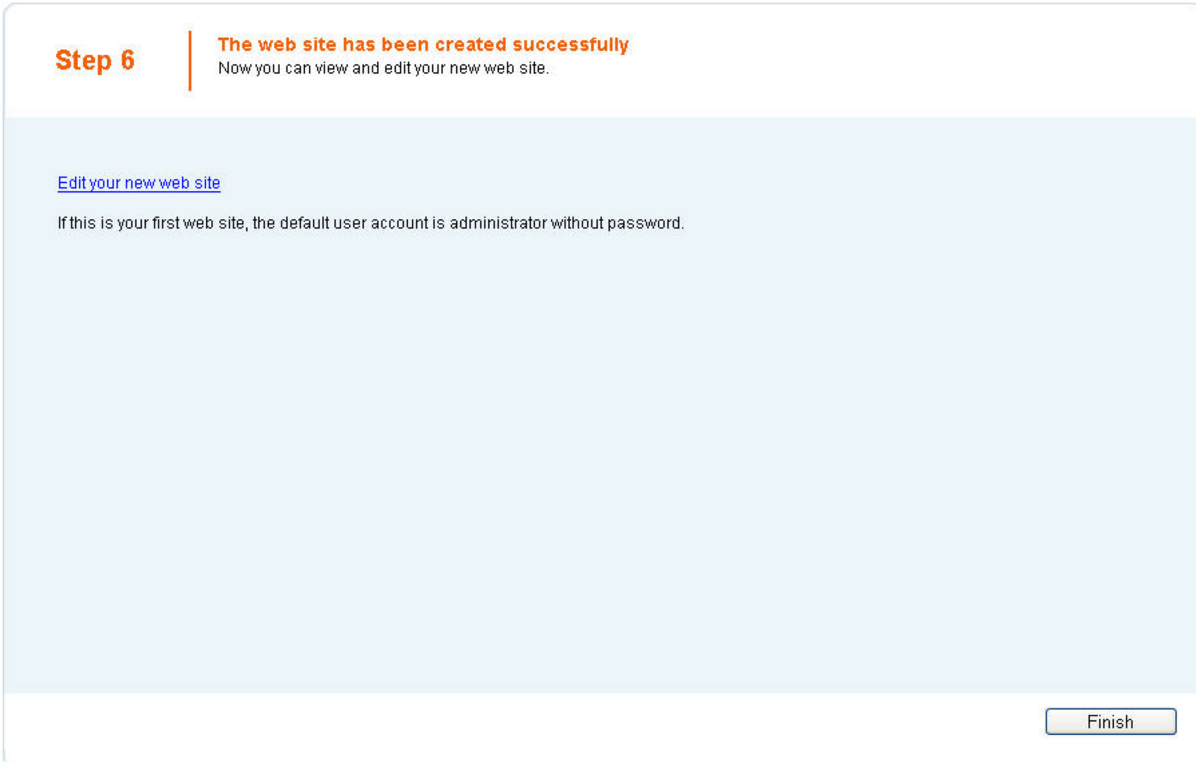
## Step 5

A log will be displayed, showing you the progress of files import.




When the export process finishes, click the **Next** button.

**Step 6**

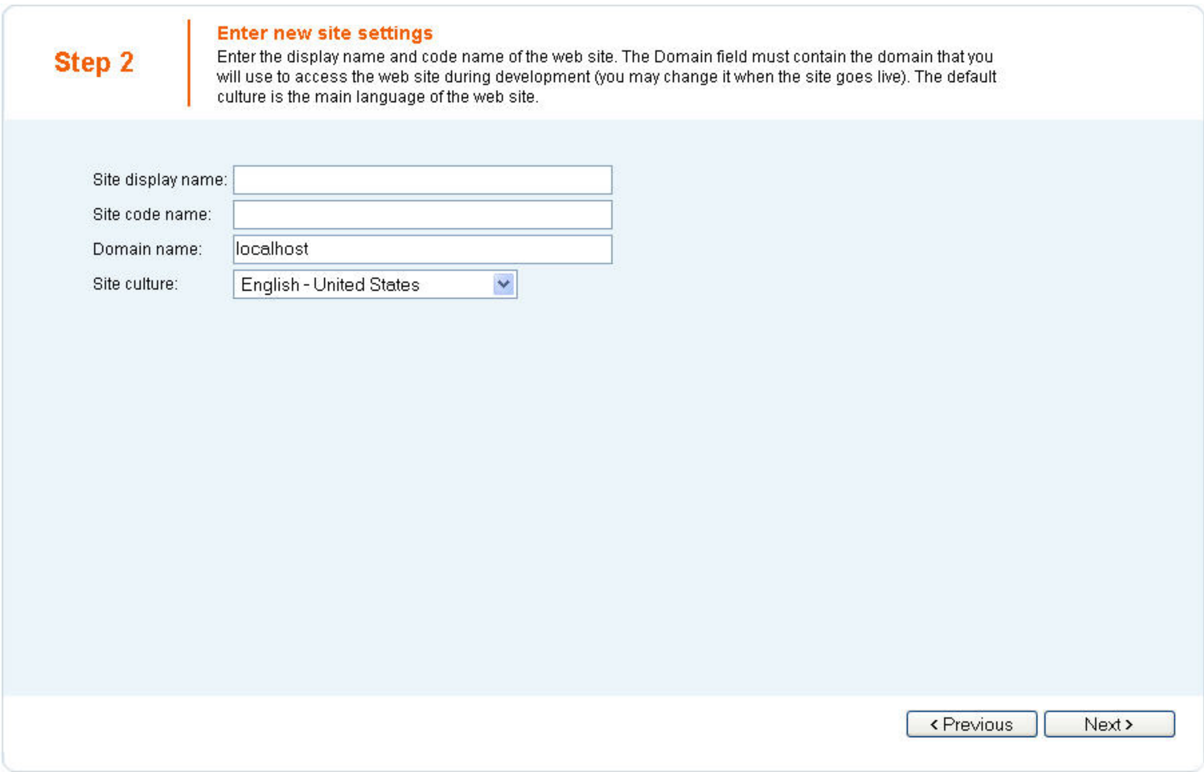



If you have reached Step 6, you have successfully created the new web site. Click the **Edit your new web site** link to switch to CMSDesk and start editing the site immediately. Alternatively, click the **Finish** button to get back to **Site manager -> Sites**.

## 'Create a new site using a wizard' version:

### Step 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.
**Site culture** - default culture of the site.



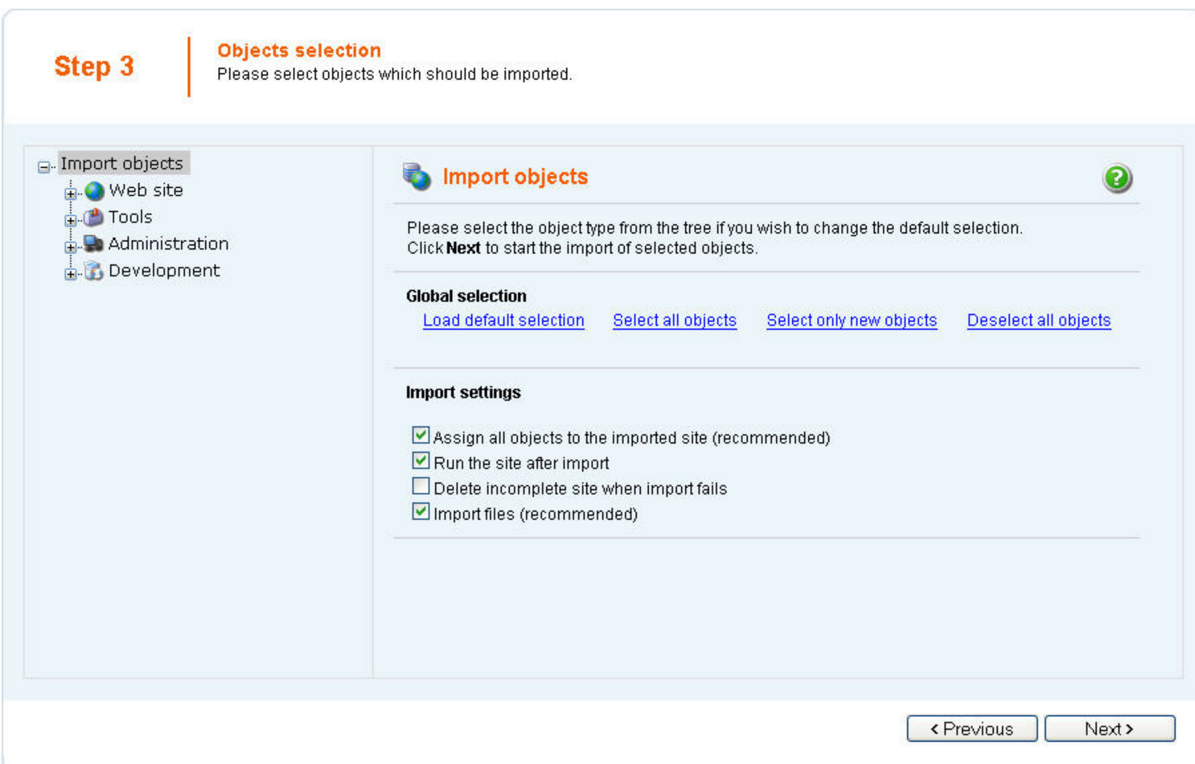

Click **Next** to continue.

**Step 3**

In Step 3, you can select which of the objects from the import package are to be imported and which are not. 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 which objects are to be exported and which are not.

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.

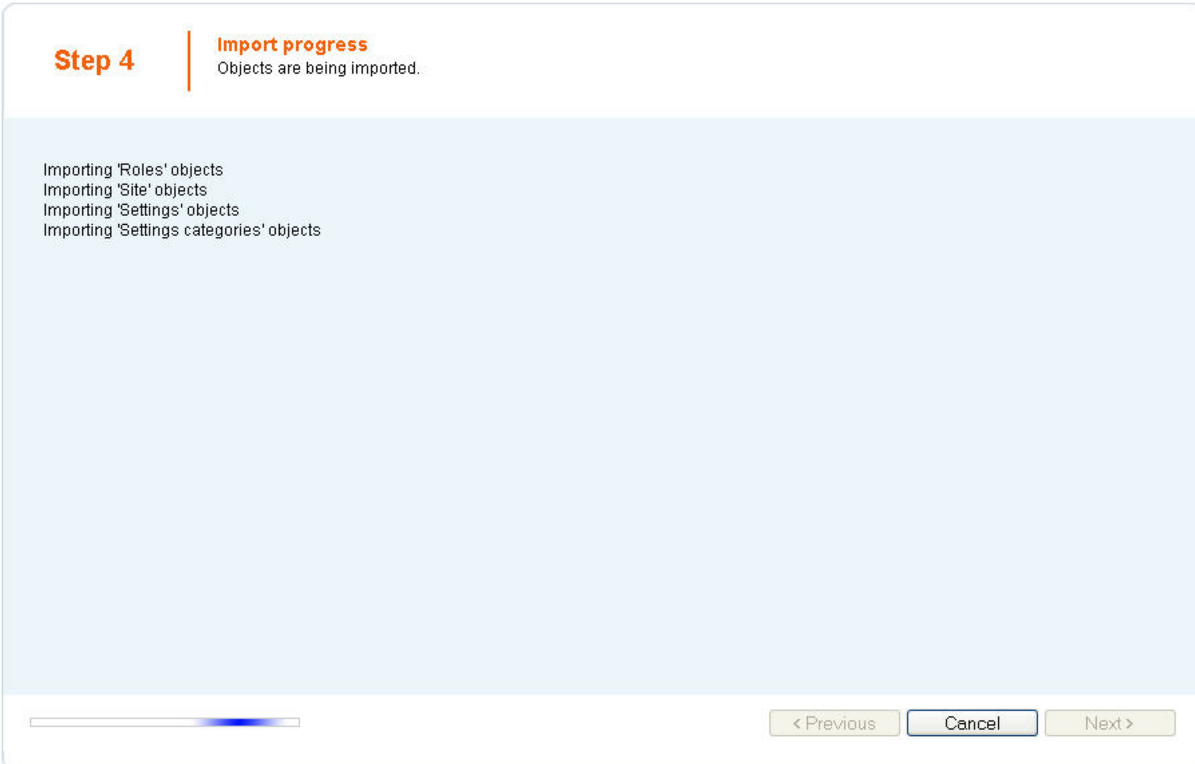
**Import settings**
- **Assign all objects to the imported site (recommended)** - if checked, all imported site related objects will be assigned to the imported site.
- **Run the site after import** - if checked, the updated site will be run after the import is finished.
- **Delete incomplete site when import fails** - if checked, incomplete site will be deleted when import fails.
- **Import files (recommended)** - if checked, files will be imported.




Click **Next** to continue.

<u>**Step 4**</u>

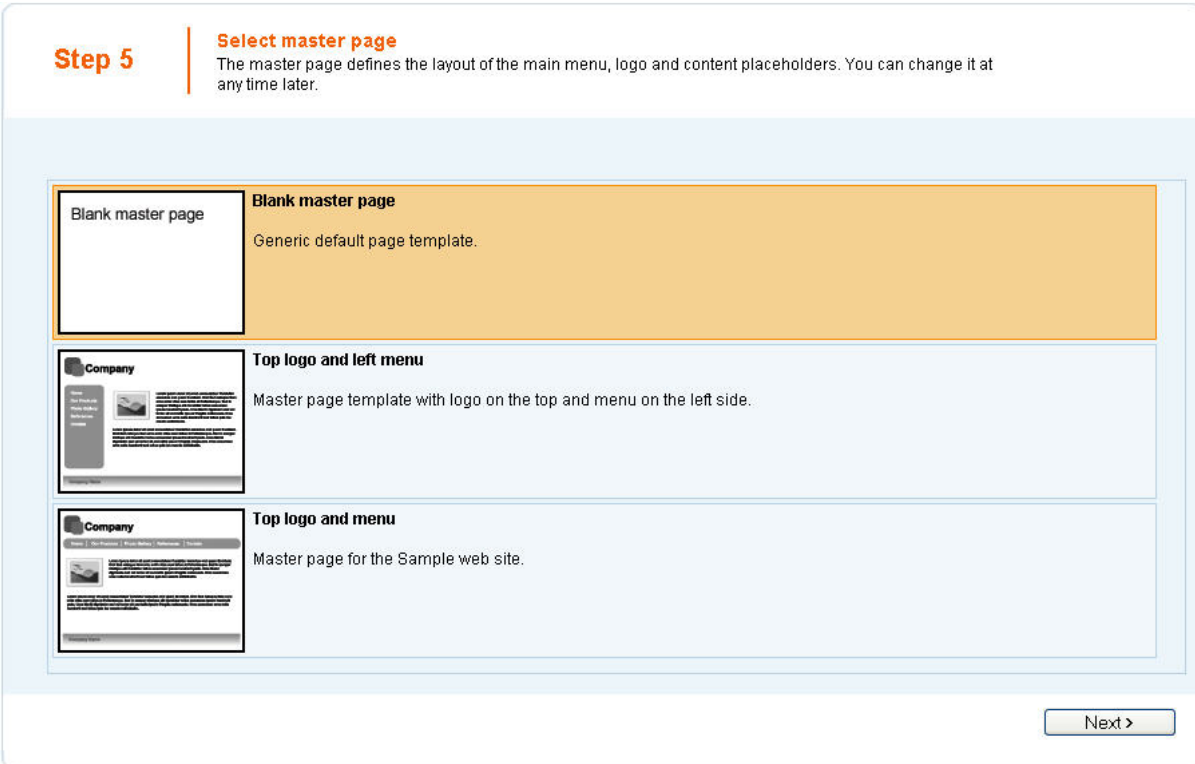
A log will be displayed, showing you the progress of files import.




When the export process finishes, click the **Next** button.

**Step 5**

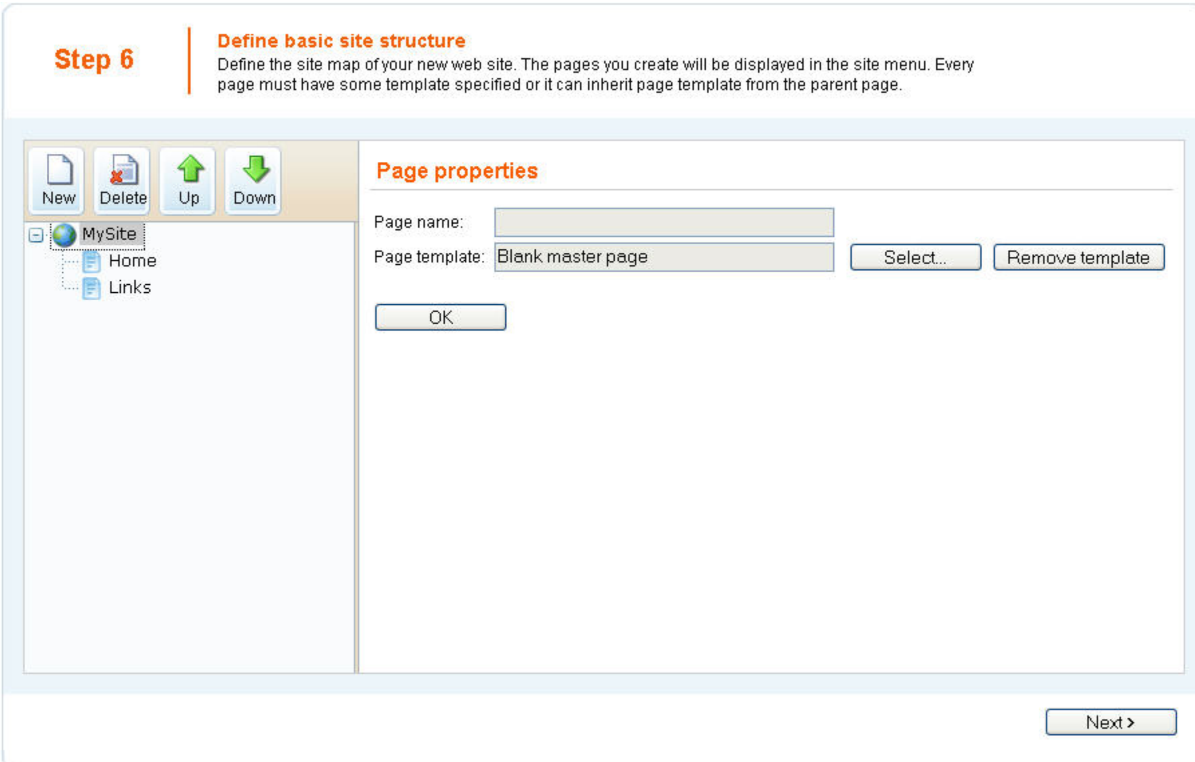
In Step 5, you can select the master page layout. This defines the basic visual structure of the website. These settings can be altered any time later, no matter which layout you have selected.




Select one of the layouts and click **Next**.

**Step 6**

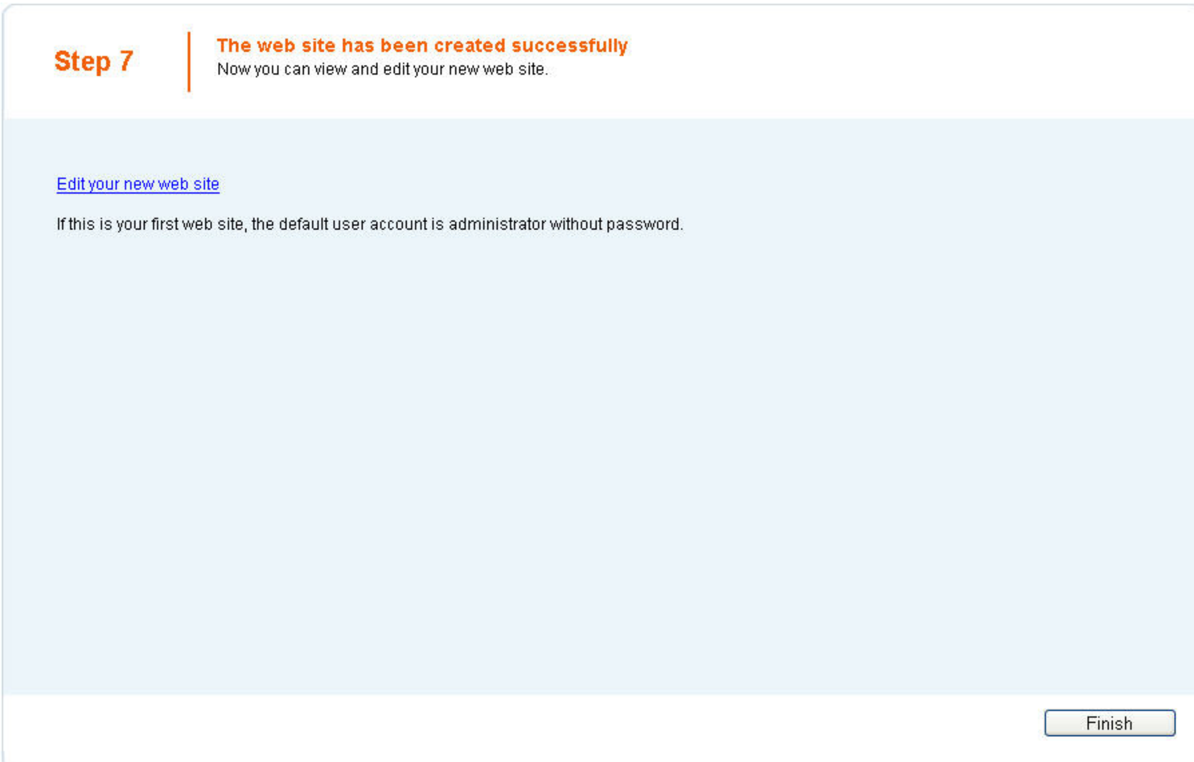
Site map of you new web site can be defined in Step 7. Select the node of the tree under that you want to place the new page and click **New**. Enter the name of the new page and select one of the page templates. Alternatively, you can inherit the page template from the parent in the tree by clicking the **Inherit template** button. Click **OK**. The newly created site will appear in the tree view.




Repeat this procedure until you have defined the desired site structure, then click **Next**.

**Step 7**

If you have reached Step 7, you have successfully created the new web site.




Click the **Edit your new web site** link to switch to CMSDesk and start editing the site immediately. Alternatively, click the **Finish** button to get back to **Site manager -> Sites**.

## 3.7 Deployment to the live server

With Kentico CMS, you can easily develop the web site on your local machine. When the web site 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 CMSSiteUtils\Export folder.

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:

- Run **Kentico CMS Web Installer on your local development computer**.
- Choose to create a new web site **on a remote (production) server** and deploy the web site into some local disk folder.
- **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 web site in a sub-folder, you need to create virtual directory manually as described in Creating a virtual directory.
- Copy your exported package into the **<web site root>\CMSSiteUtils\Import** folder.
- 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).
- Open the page /default.aspx of your live web server in the web browser.
- You will be displayed with Database setup wizard. Create the database for Kentico CMS.
- You may be asked for the license key for your production domain.
- In step 3, choose to **Import existing Kentico CMS web site**.
- Choose to import the previously exported package and overwrite all duplicates. The import wizard is described in chapter Importing a site.

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

- Go to the **Sites** section. Edit web site properties and make sure the web site domain and domain aliases are configured correctly for the production domain(s).
- 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.
- Go to the **Sites** section. Click the **Open live site** icon next to your new site and make sure the web site is displayed correctly.

### Alternative Approach

If you, for some reason, cannot deploy the web site 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 web site.
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 chapter Configuration for Medium Trust environment.

Related topics: Pre-compilation (Publish function), Configuration for Medium Trust environment

## 3.8 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:

- Install Kentico CMS on your **local machine** if you haven't done so yet.
- Run **Kentico CMS Web Installer** (you can find it in Start -> All Programs -> Kentico CMS).
- 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 2005 installed.
- **Copy the files** from your local folder C:\tempfiles to the root of the web site 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.
- **Open web browser** and navigate to your web site. You will be displayed with **Database Setup Wizard**.
- Go through the wizard and create a **new Kentico CMS database** on your live server.
- At the end of the Database Setup Wizard, choose either to create a new web site or import your

existing Kentico CMS web site.

**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

## 3.9 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:

- Install Kentico CMS on your **local machine** if you haven't done so yet.
- Run **Kentico CMS Web Installer** (you can find it in Start -> All Programs -> Kentico CMS).
- 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 2005 installed.
- **Copy the files** from your local folder C:\tempfiles to the root of the web site 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.
- 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" />
```

- **Open web browser** and navigate to your web site. You will be displayed with **Database Setup Wizard**.
- 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.
- Choose either to create a **new web site or import your existing** Kentico CMS web site.
- After you create a new web site, 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.

## 3.10 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 Web site files and create a virtual directory for it.

## 3.11 Uninstallation

To uninstall Kentico CMS from your computer, open **Start -> Control Panel -> Add/Remove Programs**. Choose Kentico CMS and click **Uninstall**.

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

## 3.12 SMTP server configuration

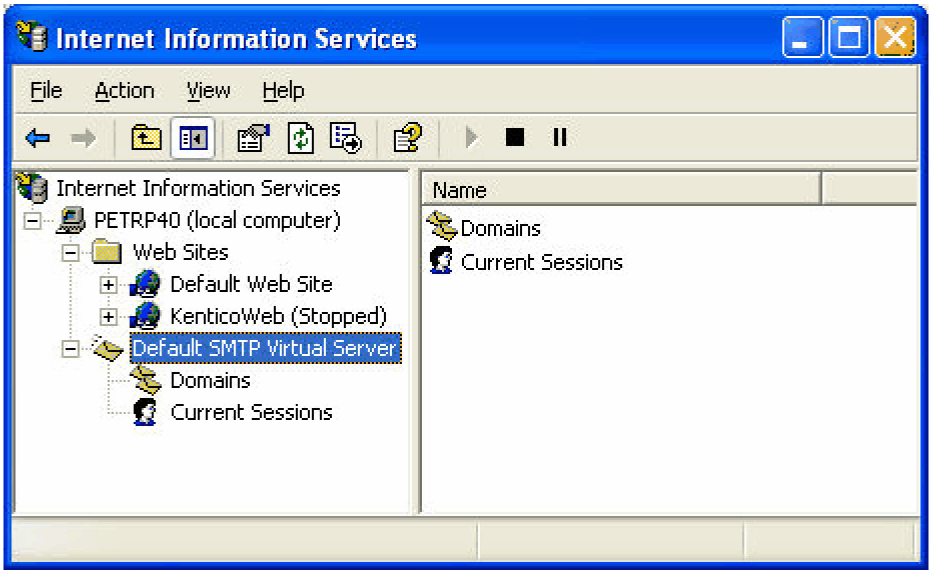
If you want to send e-mails from the web application, you may need to configure the SMTP server.

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:

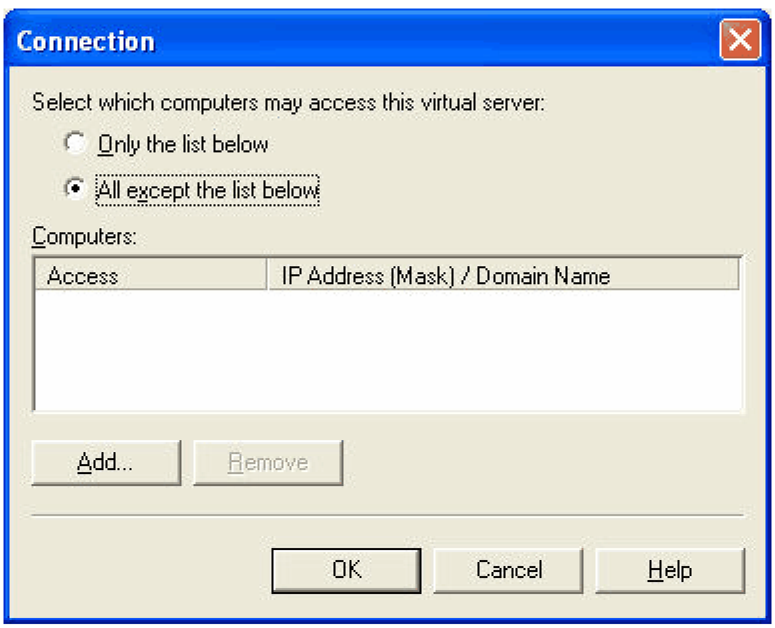
| | |
|---|---|
| SMTP server | The name, address or IP address of your SMTP server, including the port number (if it's not 25). |
| SMTP server user | User name - use this value only if your SMTP server requires authentication. |
| SMTP server password | Password - use this value only if your SMTP server 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:

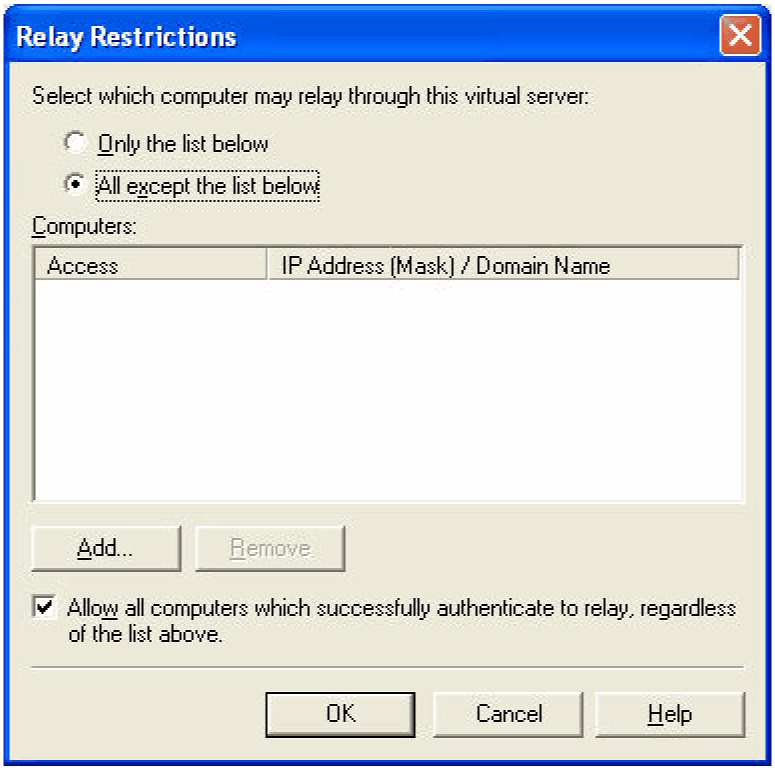
1. Open **Control Panels -> Administrative Tools -> Internet Information Services** console. Check that the SMTP server is installed and running:




2. Open SMTP server properties and choose tab **Access**. 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.

# 3.13 Additional configuration tasks

## 3.13.1 Overview

The following chapters contain description of different configuration tasks that require changes to your system. You needn't perform them for standard installation.

## 3.13.2 Configuration of extension-less URLs

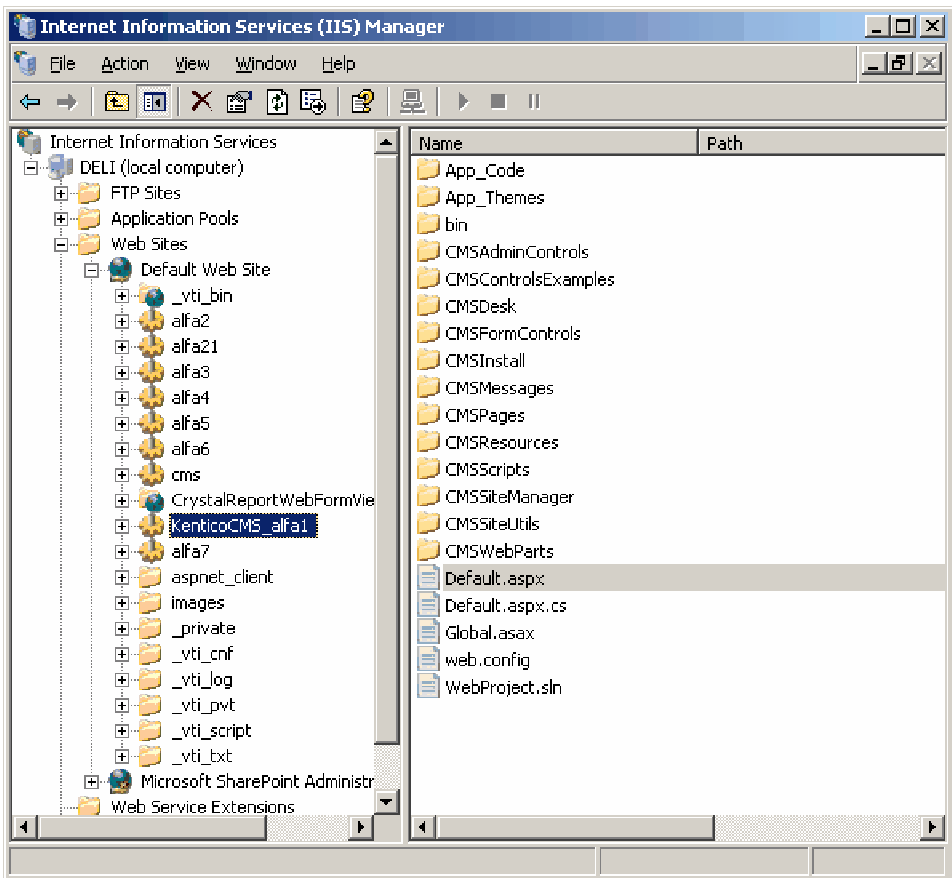
This chapter describes how to configure the system for using extension-less URLs, such as
**`http://www.mydomain.com/products/kenticocms`**
instead of
**`http://www.mydomain.com/products/kenticocms`*`.aspx`***

## Prerequisites

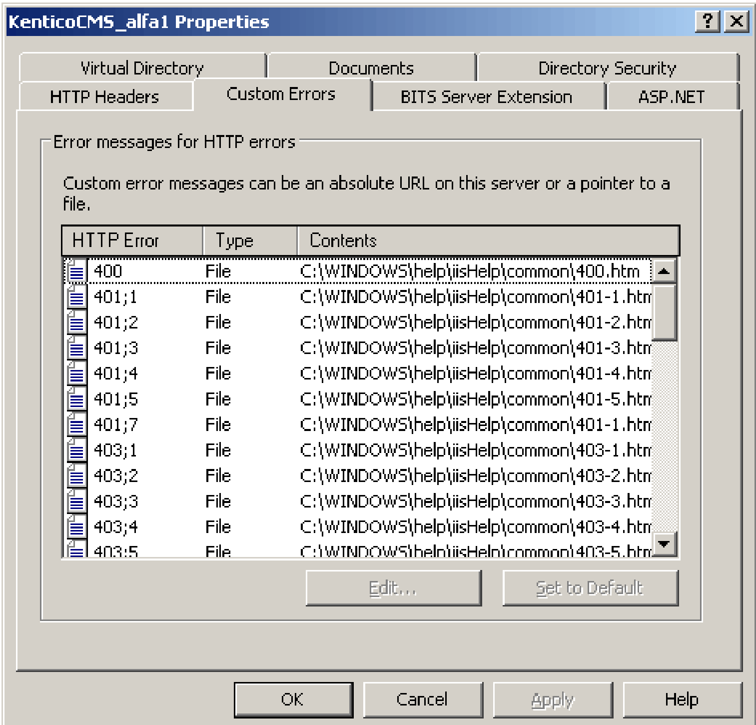
You can use this configuration on Windows XP, Windows 2000 and Windows Server 2003 with Internet Information Services (IIS) installed. It's not possible to use it with Visual Studio 2005 built-in web server.

## Configuration

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

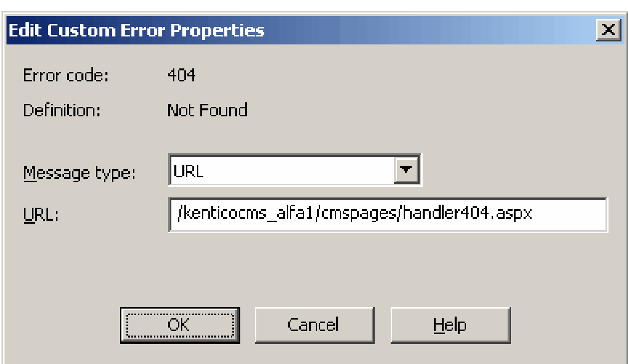


2) Right-click the directory (or web site) and choose **Properties** and then click the **Custom Errors** tab:




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.
  Example: 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.

5) Now go to **Kentico CMS Site Manager -> Settings**. Choose either the global configuration or site-specific configuration and choose the **URLs** configuration section. Clear the **Friendly URL extension** textbox value and click **Save**.

If you want to enable other friendly URL extensions along with extension-less URLs, you can enter the URLs divided by a semicolon, while two semicolons enable the extensionless URLs. The first value will be used for generating the URLs displayed in browsers.

Example: entering **;;.aspx;.html** enables .aspx and .html extensions and extensionless URLs; because the two semicolons are entered first, extensionless urls will be displayed in your browser.

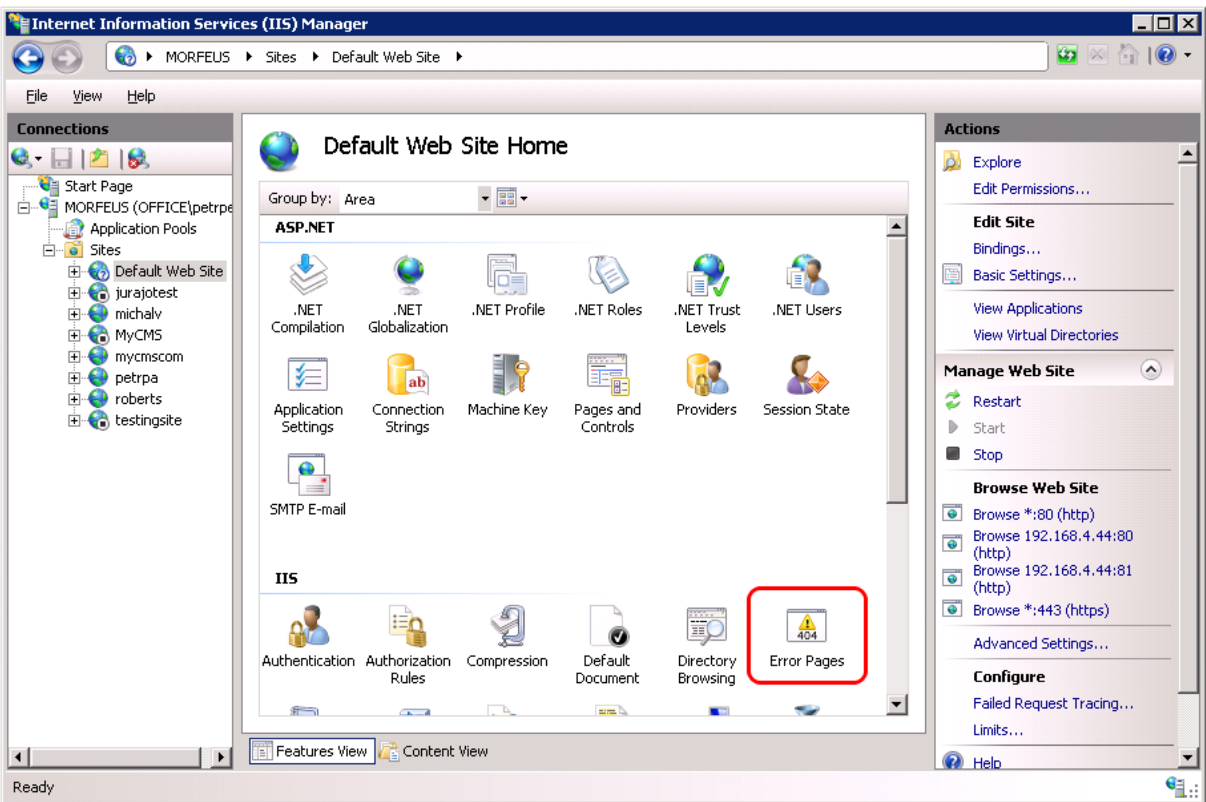
Now when you go to the live web site, you will see that all URLs in menus and listings are rendered without extension. You may, however, need to update some static links that were created with default .aspx extension.

## Configuring extension-less URLs on IIS 7

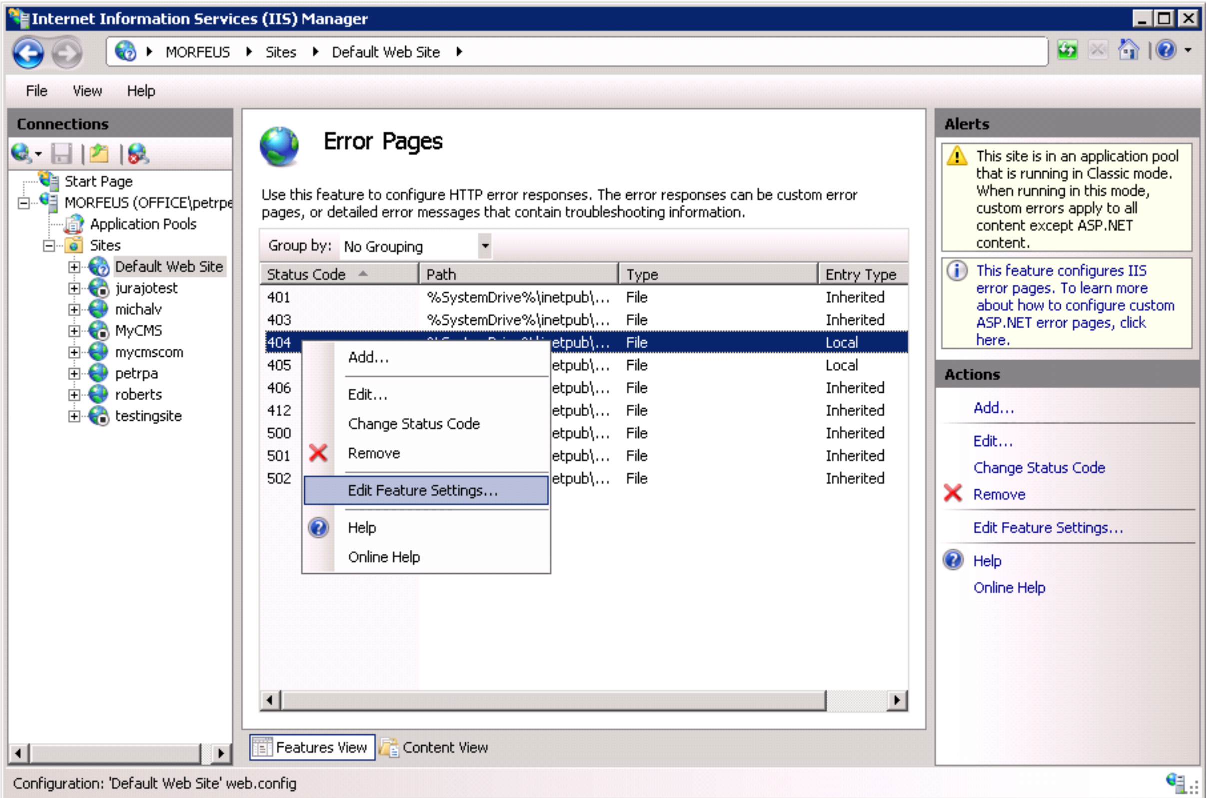
The setup procedure on IIS 7 is slightly different due to IIS 7's different user interface.

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

2) Select your web site from the tree on the left and open the **Error pages** section.



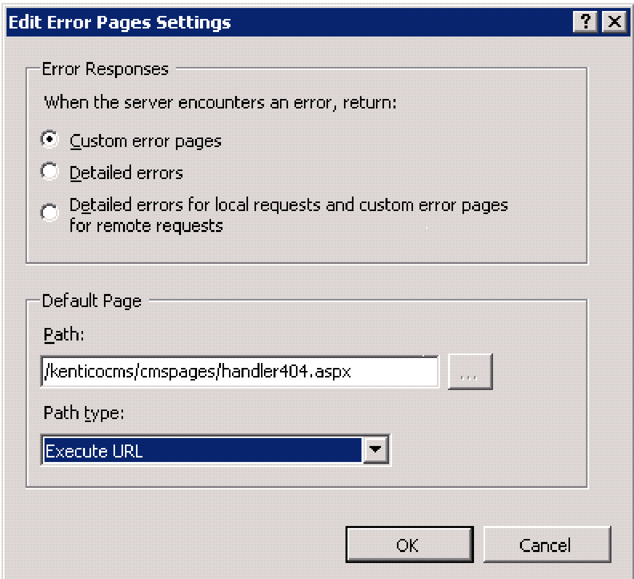
3) Right-click the **404 error** and choose **Edit Feature Settings**.



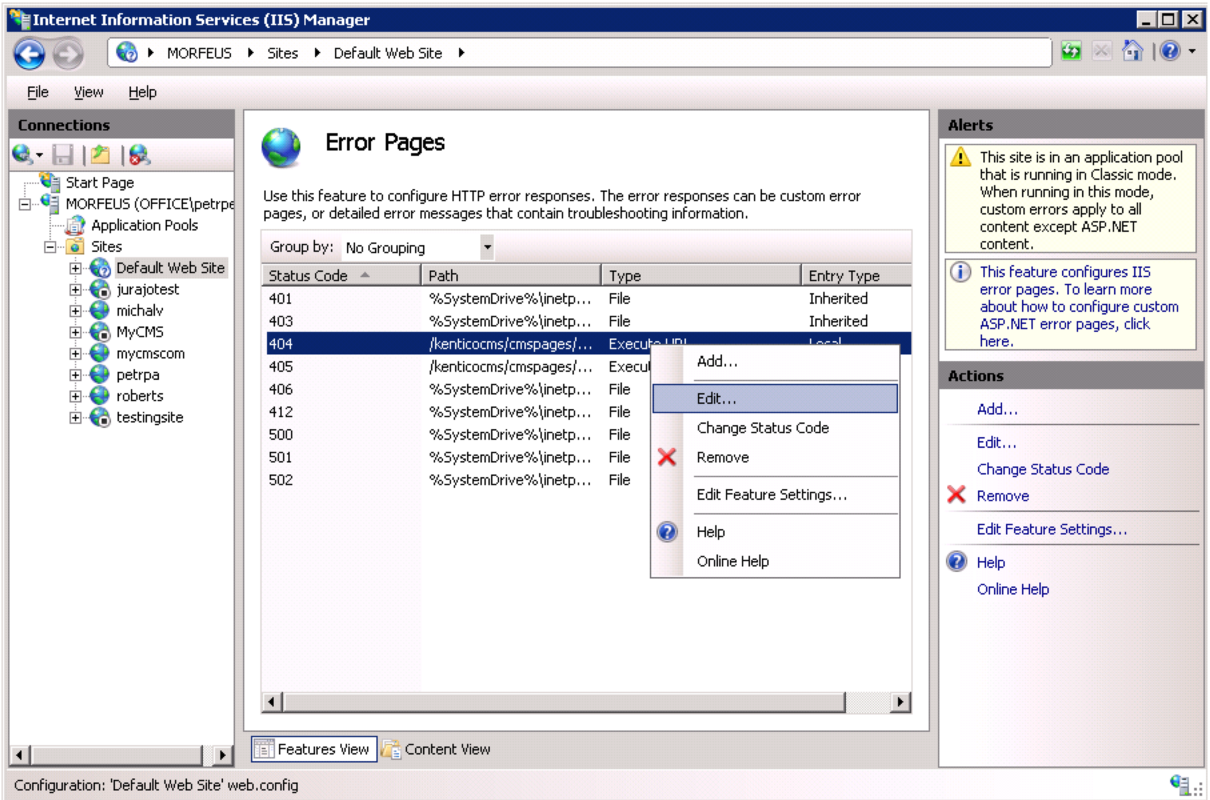
4) Enter the following values:

- **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

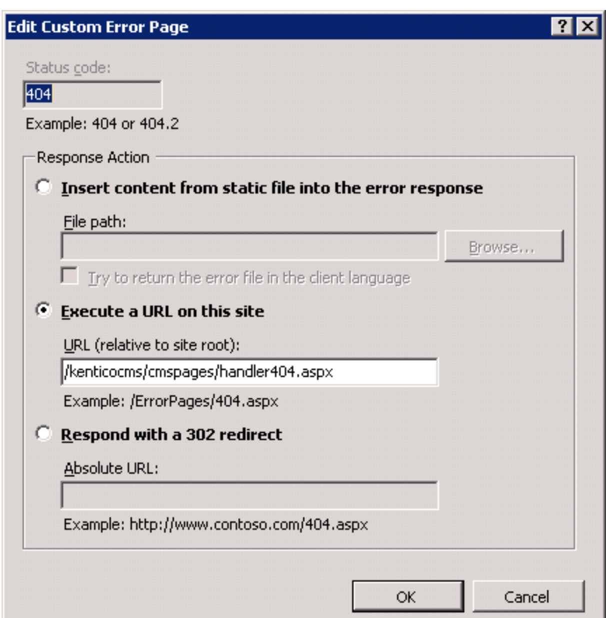
Click **OK**.



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.

9) Now go to **Kentico CMS Site Manager -> Settings**. Choose either the global configuration or site-specific configuration and choose the **URLs** configuration section. Clear the **Friendly URL extension** textbox value and click **Save**.

If you want to enable other friendly URL extensions along with extension-less URLs, you can enter the URLs divided by a semicolon, while two semicolons enable the extensionless URLs. The first value will be used for rendering the URLs displayed in browsers.

Example: entering **;;.aspx;.html** enables .aspx and .html extensions and extensionless URLs; because the two semicolons are entered first, extensionless urls will be rendered in your browser.

Now when you go to the live web site, you will see that all URLs in menus and listings are rendered without extension. You may, however, need to update some static links that were created with default .aspx extension.

### 3.13.3 Configuration of custom URL extensions (.html or other)

The procedure is the same as for extension-less URLs. You will only configure a different extension or set of extensions at the end of the procedure:

Go to **Site Manager -> Settings -> select Global or particular web site -> URLs** -> set **Friendly URL Extension** value to **.html** and click **Save**.

Now when you go to the live web site, you will see that all URLs in menus and listings are rendered with .html extension. You may, however, need to update some static links that were created with 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**

Extensions are divided by a semicolon. The first extension is used as the default one and the links will be generated with it. 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 will also be supported.

## 3.13.4 Configuration for Medium Trust environment

This chapter describes the procedure to run the Kentico CMS in the 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 web sites running on the server. Since version 2.2, 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 with this trust level:

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

### Virtual path provider

This library provides the interface to the 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 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 document transformation files
- **~/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 web site 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 not that **you shouldn't run the system in medium trust while developing the web site**. You should use this trust level only for the live web site.

### 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:

- Go to **Control panel -> Administrative tools -> Microsoft .NET Framework 2.0 Configuration**
- Select the **Assembly cache**, click on **Add an Assembly to the Assembly Cache** and select the library file **bin\Microsoft.Web.Services3.dll** from your web project.
- Delete the file **bin\Microsoft.Web.Services3.dll** from your web project if it's present.
- 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 web site

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

If your web site 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 file **bin/Microsoft.Web.Services3.dll** 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.

## 3.13.5 Creating a virtual directory

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

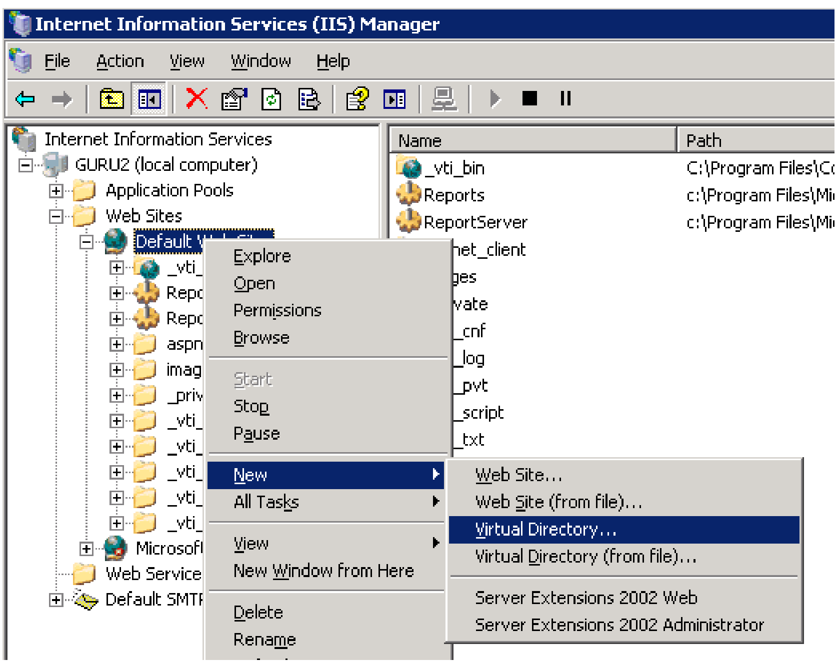
**Application root**

The root of the web site 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**.

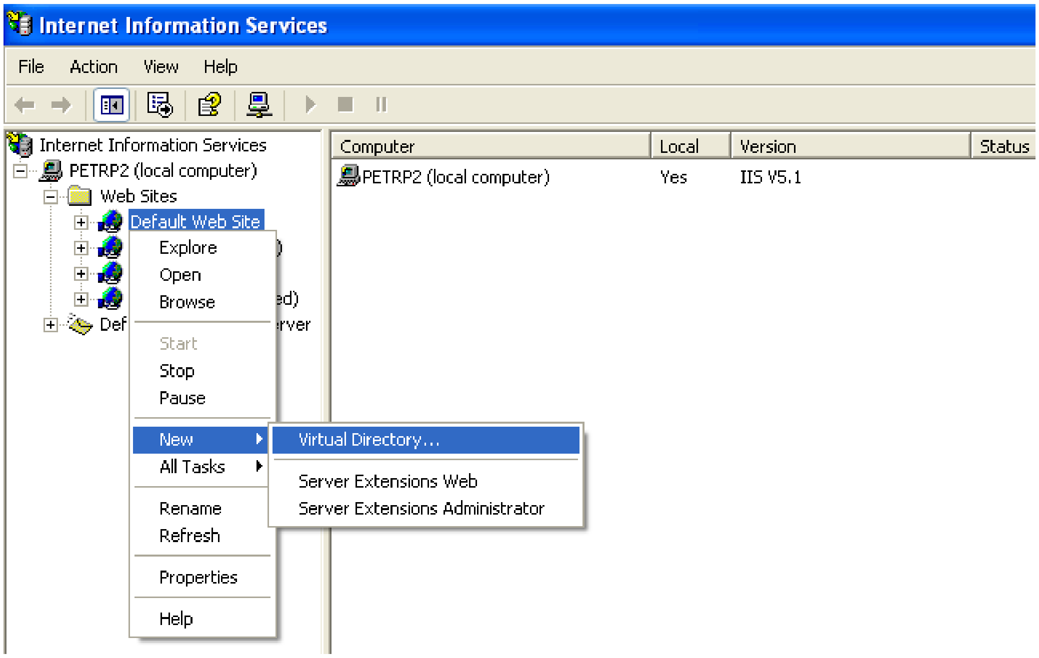
Follow these steps to create and configure a virtual directory on your web server:

### Configuring a Virtual Directory

On Windows Server 2000/2003, open **Start -> Administrative Tools -> Internet Information Services (IIS) Manager**. Right-click **Computer -> Web Sites -> chosen web site** and choose **New -> Virtual Directory**.



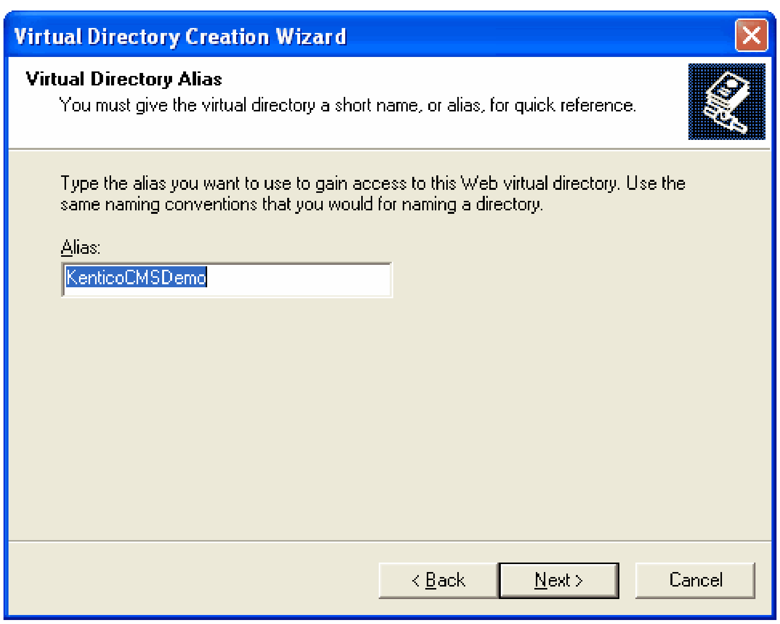
On Windows XP, **open Start -> Control Panel -> Administrative Tools -> Internet Information Services**.Right-click the **Computer -> Web Sites -> Default Web Site** item (or other web site if you're running multiple web sites on the same computer) and choose **New -> Virtual Directory...**.



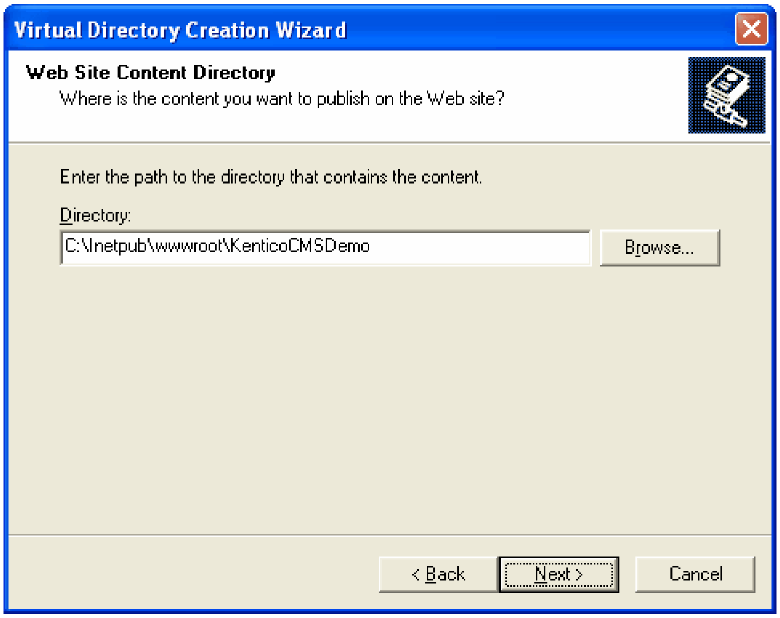

The rest is same for all operating systems:

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

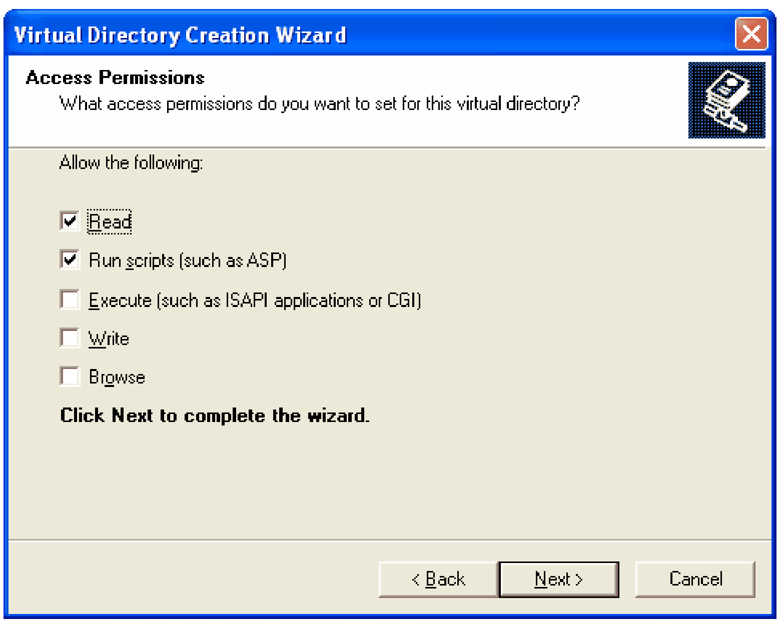
2. Enter the **Virtual Directory Alias**. If you want the web site to run as http://localhost/KenticoCMSDemo, you need to enter alias **KenticoCMSDemo**. Click **Next**.



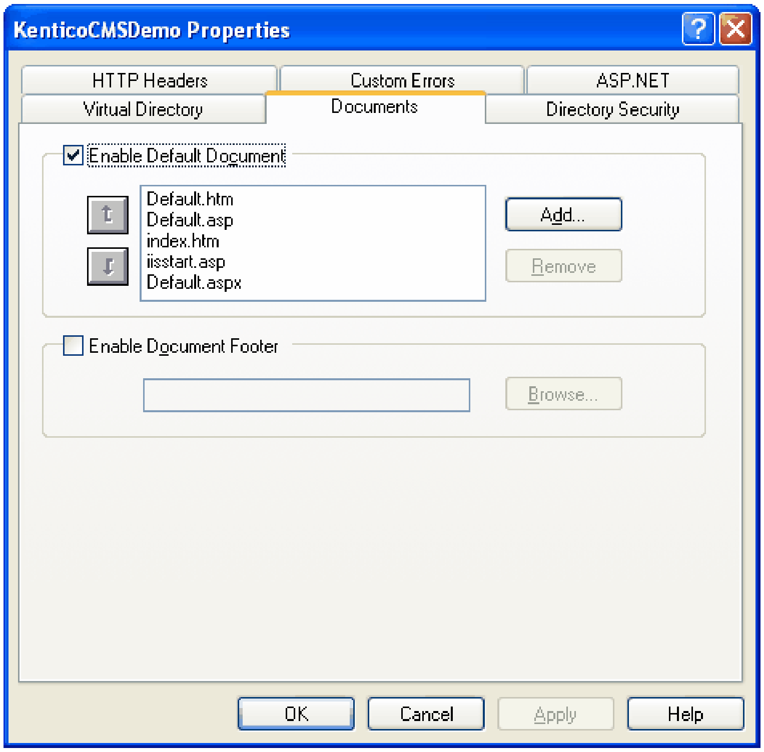
3. Now choose the disk folder on your machine where the Kentico CMS web project files are placed. Please note that you cannot use a remote (shared) disk. The web project files are NOT the files in the C:\Program Files folder - you need to create the web project using the Kentico CMS Web Installer first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

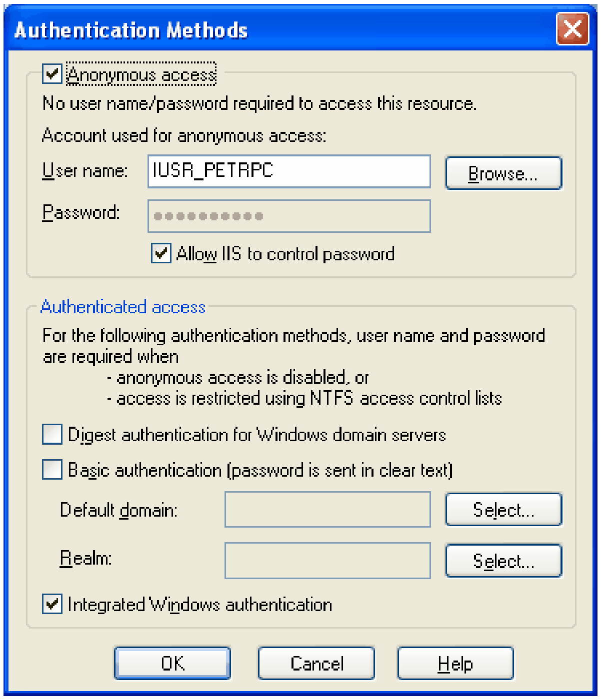


5. Right-click the newly created virtual directory and choose **Properties**. Choose the **Documents** tab and make sure the **Enable Default Document** box is checked and the list contains the **Default.aspx** file.




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.

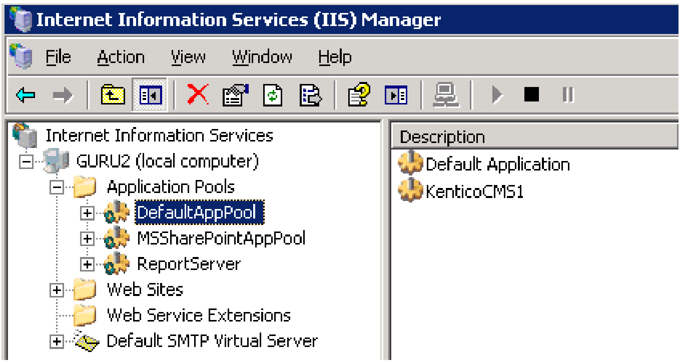
7. Choose the **Directory Security** tab, click **Edit** in the **Anonymous access and authentication control** section and make sure that both **Anonymous access** and **Integrated Windows authentication** are chosen.




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

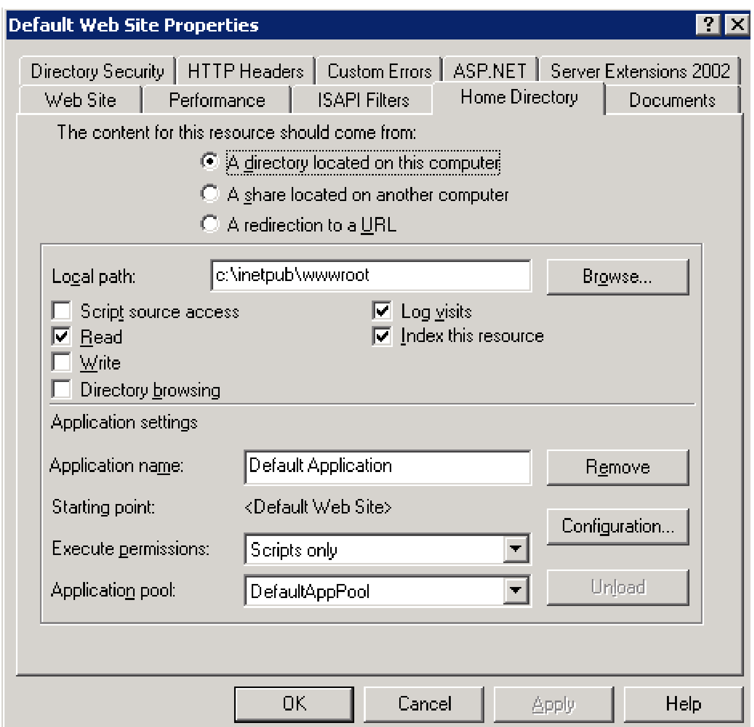
## 3.13.6 Configuring Application Pools

Application Pools provide you with additional level of web site management. They are supported only on Windows Server 2000/2003. You can configure them in **Start -> Administrative Tools -> Internet Information Services (IIS) Manager -> Computer -> Application Pools.**

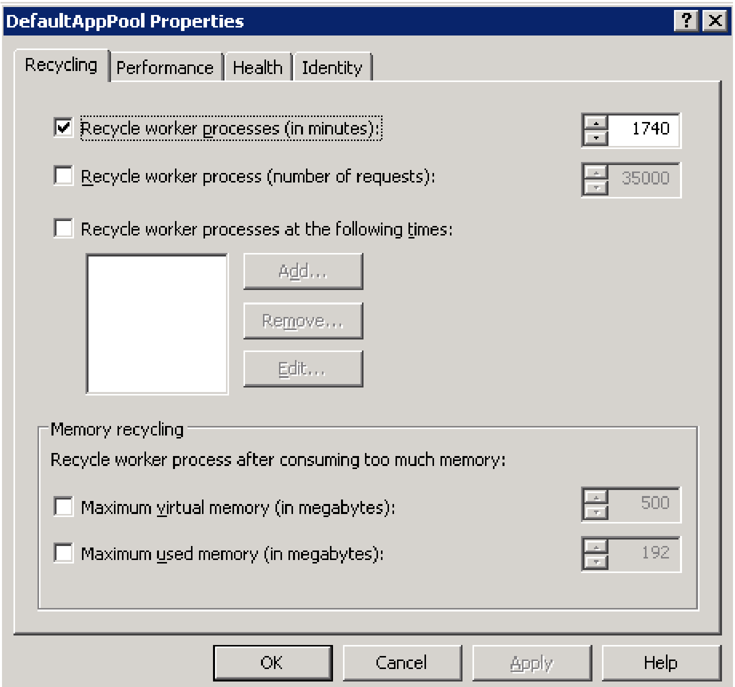



The application pools allow you to group applications (web sites) into pools that share server resources using the same rules. This chapter contains recommendations on how to configure the web site application pool for Kentico CMS.

You can check the name of the application pool in the Web Site 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 web sites, the system may behave unpredictably.

2. If you need to run multiple web sites 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 web site freezes** time to time, you can temporarily set the value to 30 minutes to ensure that the web site 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 web site 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 web site 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.

### 3.13.7 Setting up html extension for IIS7

For setting up html extensions for IIS7, please go through the following procedure

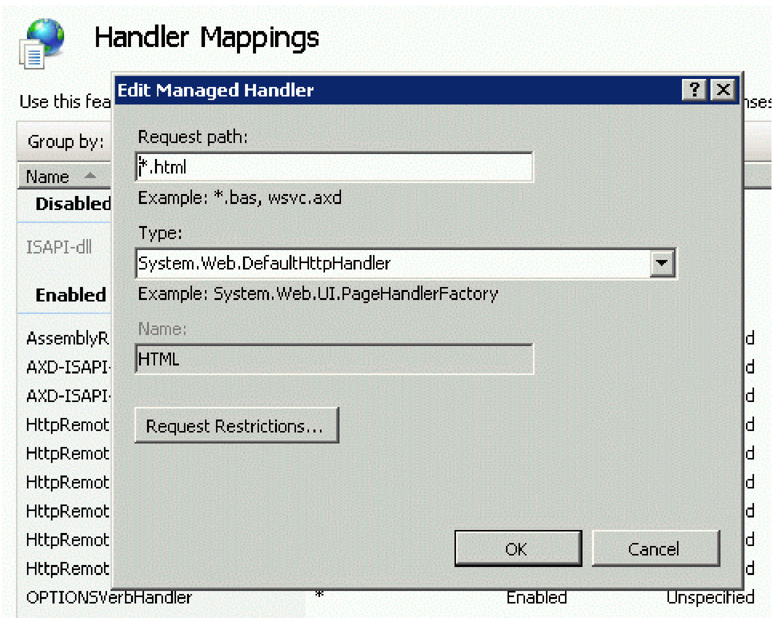
1. Run **IIS Manager** .

2. Select your web.

3. Open handler mappings.

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

5. Set dialog values just as the following image shows.



# 3.14 Configuration of full-text search in files

## 3.14.1 Configuration of full-text search in files

Kentico CMS allows you to full-text search files uploaded into 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 2000, you need to use Microsoft SQL Server 2000 Standard or Enterprise Edition - Full-text search is not supported with MSDE or in the Developer Edition.

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 web site must be configured for storing files in database (**Site Manager -> Settings -> Files -> Store files in database** must be checked).

Now follow one of the following guides:

- Configuration of full-text search on Microsoft SQL Server 2000
- 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 chapter Searching PDF files.

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).

## 3.14.2 Configuration on MSSQL 2000

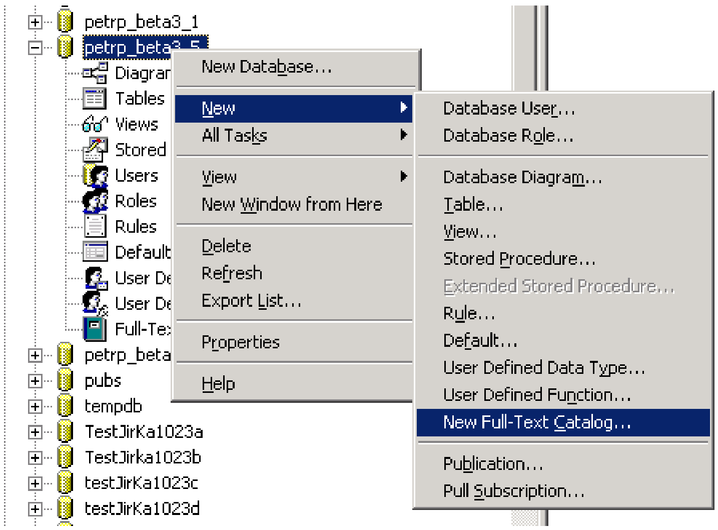
### Prerequisites

If you are using Microsoft SQL Server 2000, you need to use Microsoft SQL Server 2000 Standard or Enterprise Edition - Full-text search is not supported with MSDE or in the Developer Edition. Full-text search must be installed on your SQL Server.

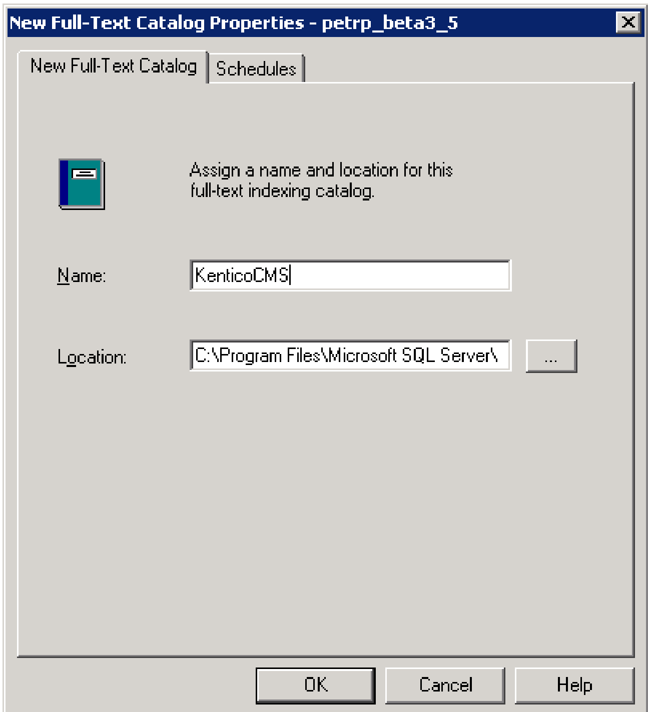
### How to set up full-text search

1. Run **Microsoft SQL Server Enterprise Manager** console.

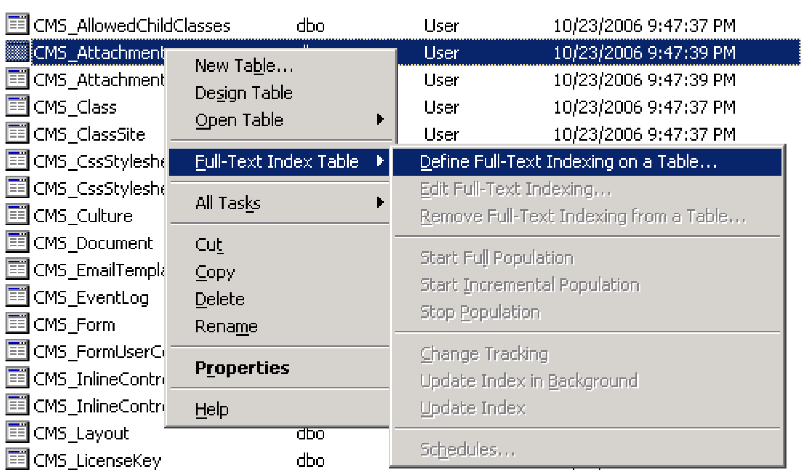
2. Locate your Kentico CMS database, right-click it and choose **New -> New Full-Text Catalog**:



3. **New Full-Text Catalog Properties** dialog appears. Enter the name of the new catalog - e.g. KenticoCMS. Click **OK** to save.

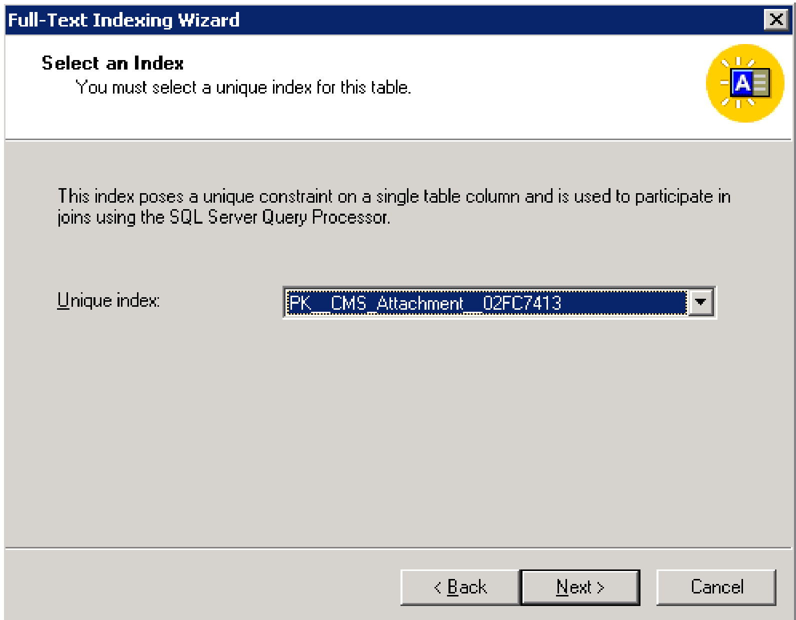



4. Right-click the **CMS_Attachment** table in the Kentico CMS database and choose **Full-Text Index Table -> Define Full-Text Indexing on a Table...**



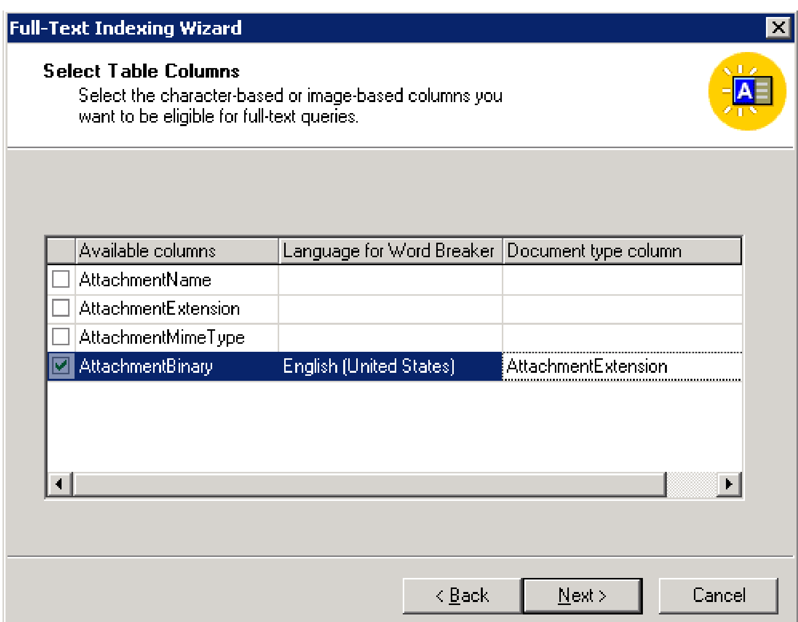

5. The **SQL Server Full-Text Indexing Wizard** opens. Click **Next**.

6. Choose the primary key PK_CMS_Attachment_... in the **Unique index** field.

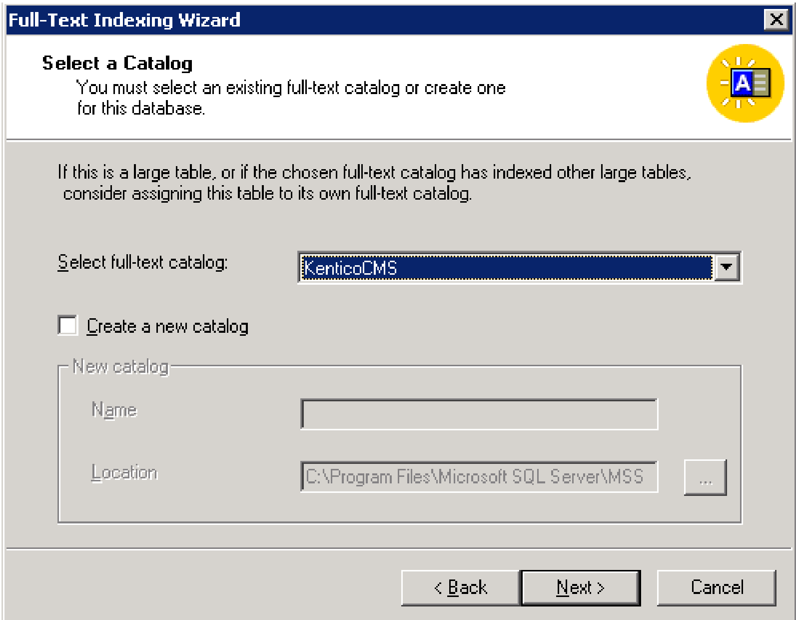



7. In the next step, check the **AttachmentBinary** column, choose the appropriate language that will be used for indexing and choose **AttachmentExtension** column as **Document type** column. Click **Next**.

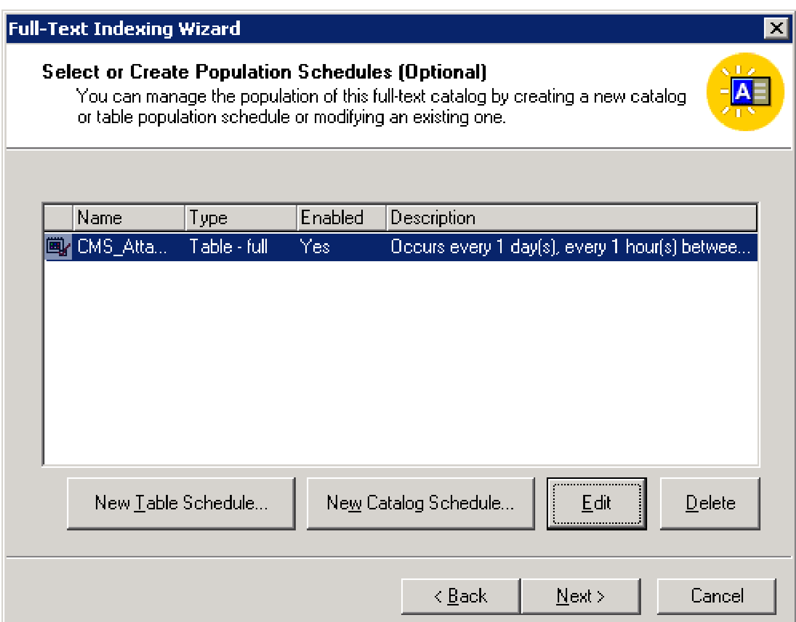
**Please note:** if you choose English language, French (Italian, etc.) documents will be indexed too, but without native support (noise words, stop words, etc.) which can lead to inappropriate search results in some cases.



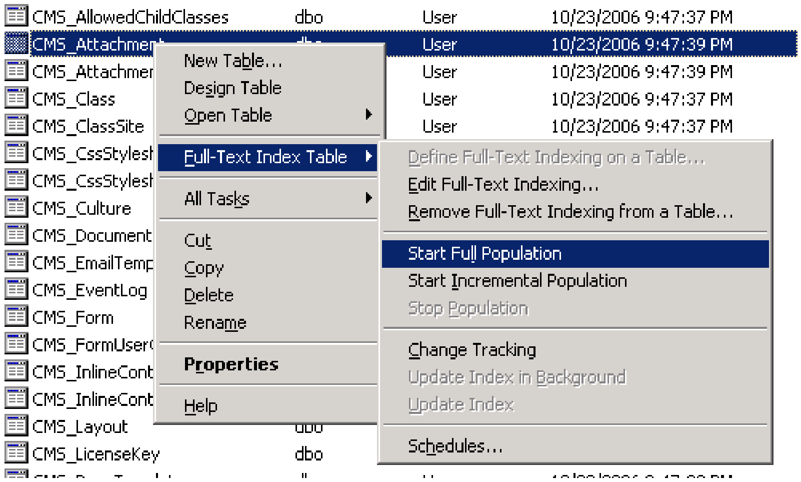
8. In the next step, choose the catalog you created in step 3. Click **Next**.



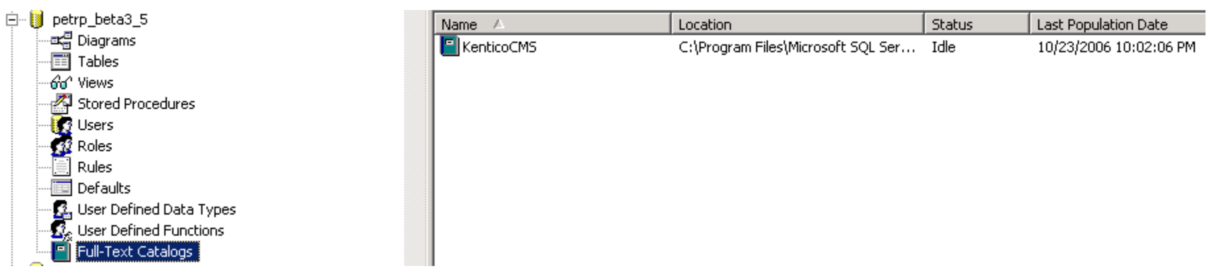

9. Now you can define a new indexing schedule. This is important, since all documents must be indexed before they can be searched. The index catalog must be updated after new files are added. You can do that manually or (preferably) you can schedule automatic indexing (incremental indexing is usually sufficient).



10. Click **Next** and click **Finish**. Now you have created a new catalog. However, it's not indexed yet. You need to start the first full indexing manually. Right-click the **CMS_Attachment** table, choose **Full-Text Index Table -> Start Full Population**.




11. Wait a few seconds and then refresh the **Full-Text Catalogs** listing. Check that the **Last Population Date** has been updated.



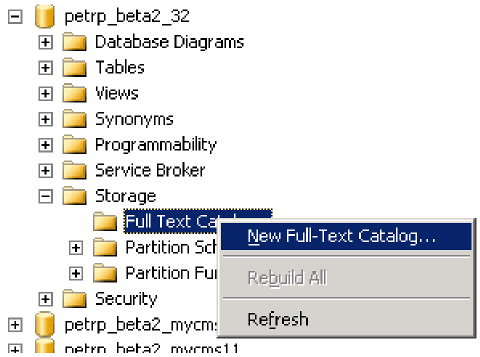

12. Sign in as administrator and go to **Site Manager -> Development -> Document types -> Root** and edit the **searchattachments** query. Uncomment the following query:

```
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 )
```

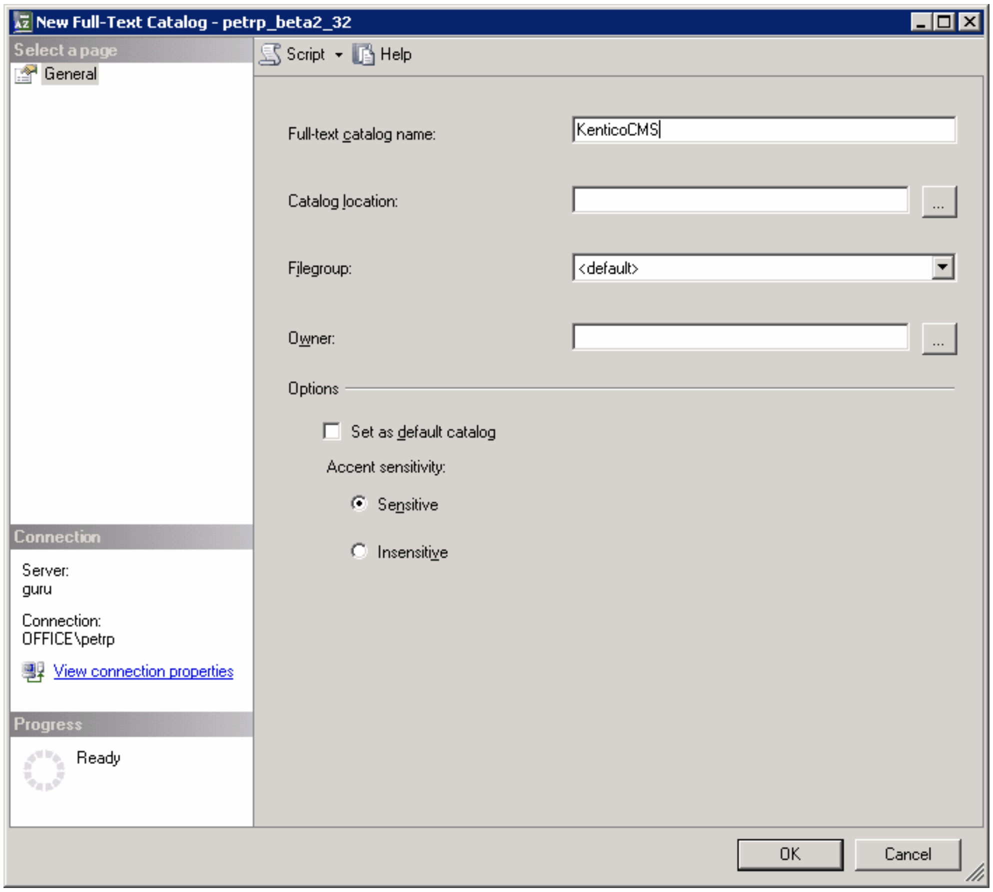
## 3.14.3 Configuration on MSSQL 2005 and 2008

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**.

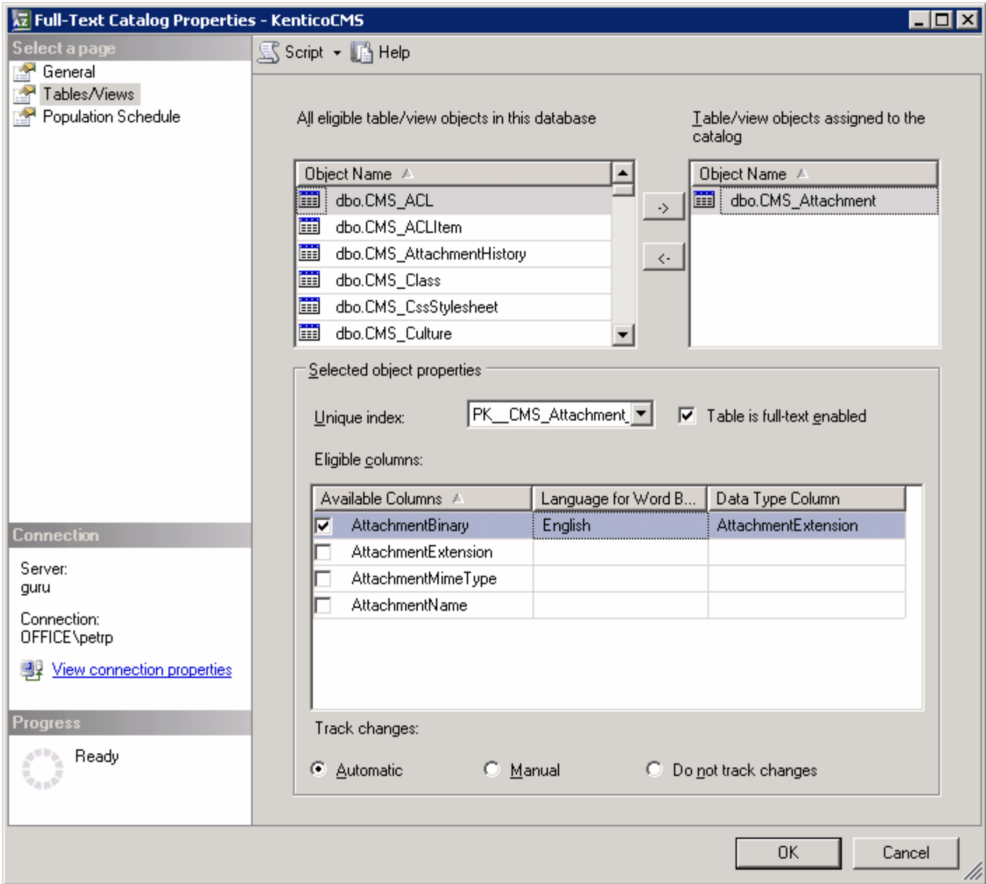



3. On the **New Full-Text Catalog** dialog, enter Full-text catalog name *KenticoCMS* and click **OK**.



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** 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 )
```

### 3.14.4 Configuration using SQL on MSSQL 2005 Express Edition

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

```
-- Activate Full text Searching on PDF on Microsoft SQL 2005
sp_fulltext_service 'verify_signature', 0
sp_fulltext_service 'load_os_resources',1

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

-- Add Full Text catalog to table
sp_fulltext_table
'CMS_Attachment','create','MyKenticoCMSCatalog','PK_CMS_Attachment_02FC7413'

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

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

**Please note:** Similarly to configuration on MSSLQ 2005, you need to go to **Site Manager -> Development -> Document types -> Root** -> **<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 )
```

### 3.14.5 Searching PDF files

Please download the free Adobe PDF IFilter 6.0 from the Adobe Web site:
http://www.adobe.com/support/downloads/detail.jsp?ftpID=2611

Run the installer and follow the instructions. 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.

**Note 1:** Adobe PDF IFilter is not a product of Kentico Software and we cannot guarantee its functionality.

**Note 2:** 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

# 3.15 Troubleshooting installation issues

## 3.15.1 Troubleshooting overview

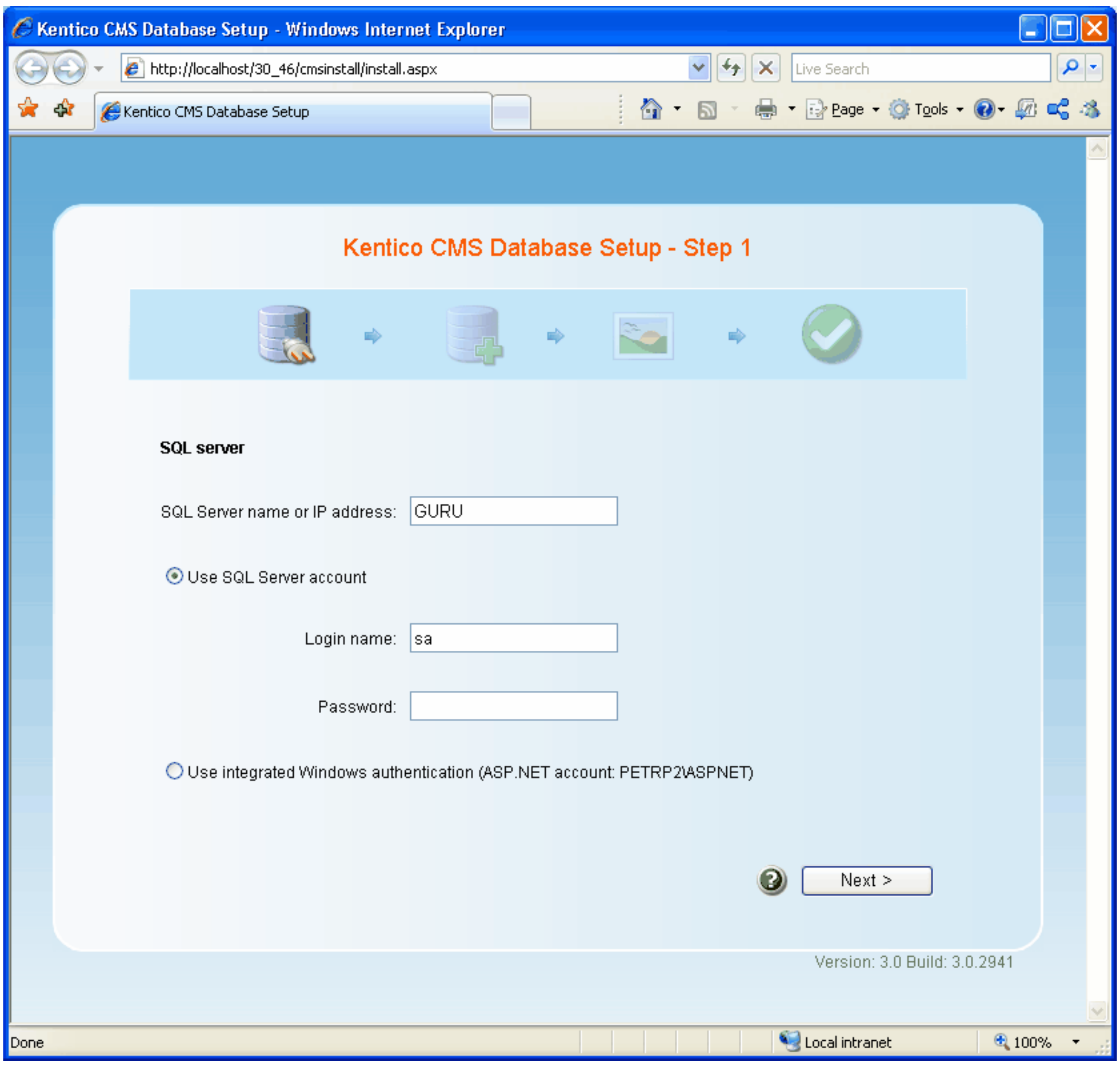
You may encounter various issues during the installation. The following chapters may help you sort them out.

You may encounter problems in following areas:

- SQL Server connection problems
- Disk permissions problems
- ASP.NET not working on Windows Server 2003

## 3.15.2 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)

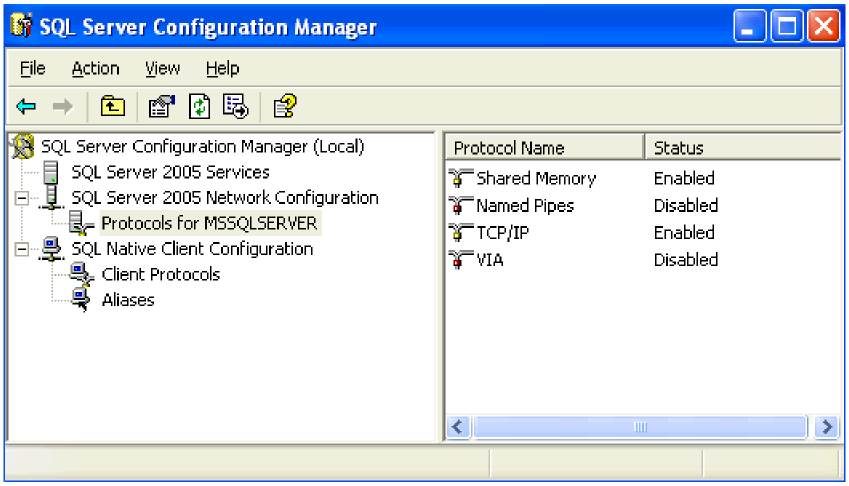
2. Make sure the server has Microsoft SQL Server 2000 or 2005 installed and running.

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 ****).

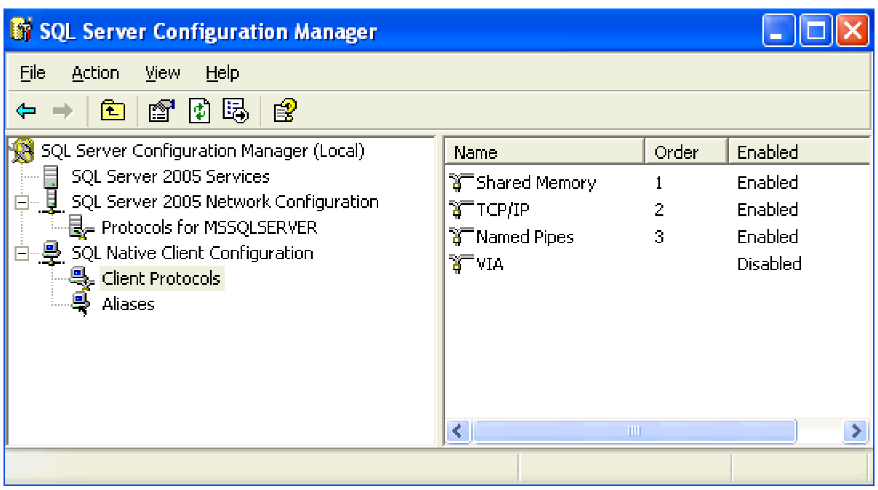
4. If you're using Microsoft SQL Server Express 2005 with default installation settings, the correct server name is **.\sqlexpress** or **computername\sqlexpress** .

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:

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

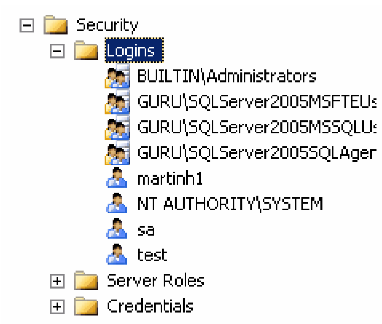
## Error 2: Login failed for user 'xy'

Error message:

| |
|---|
| Login failed for user 'xy' |

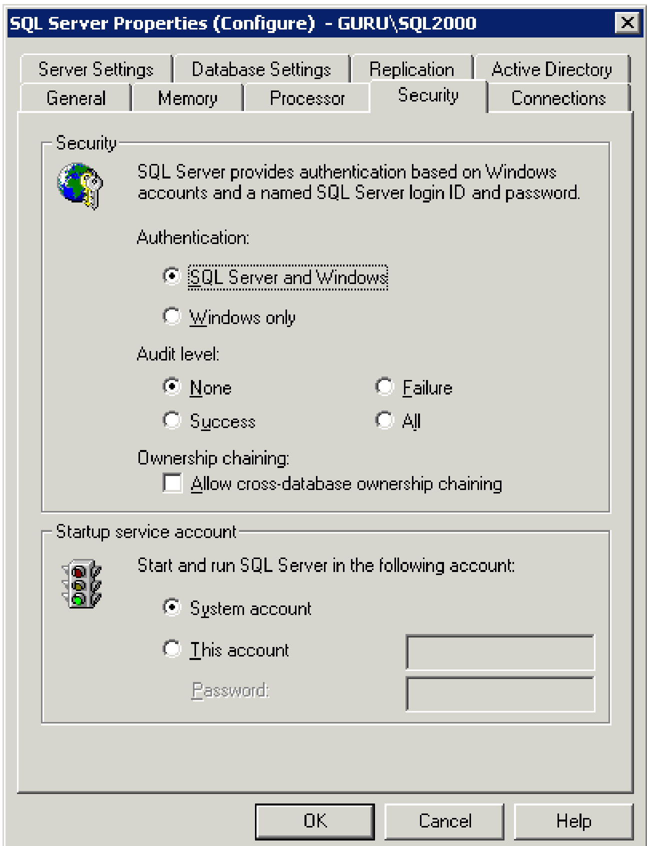
### Troubleshooting for SQL Server account

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 granted with permission to connect to the server. You can check the user account in Enterprise Manager/SQL Server Management Studio -> Server -> Security -> Logins:

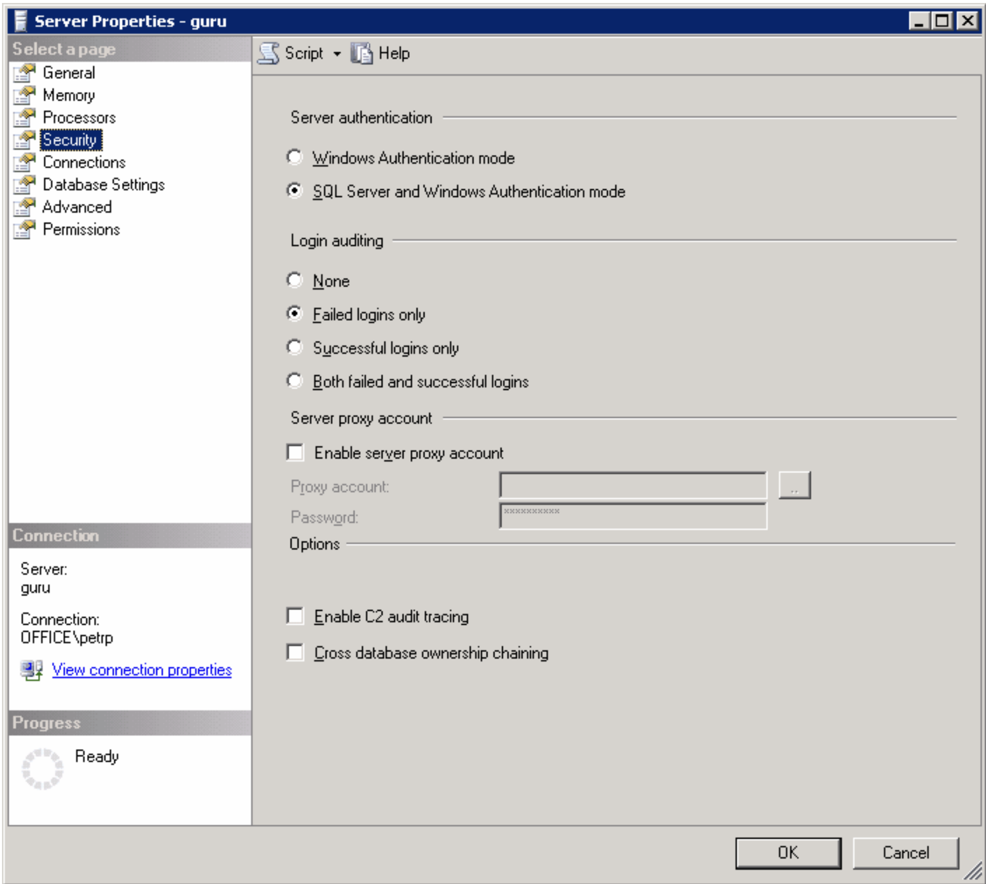


Also, check the **Server Properties -> Security** dialog in Enterprise Manager/SQL Server Management Studio and make sure your server supports **SQL Server and Windows Authentication mode**:

- SQL Server 2000:



- SQL Server 2005:




### 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. The 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 it with appropriate permissions on the server. 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.

## 3.15.3 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 an error message from the system saying that the web application cannot write to disk, you need to grant the appropriate user account with Modify permissions on the whole web site.

### User account of the web application

The web application runs under user account that depends on your environment:

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

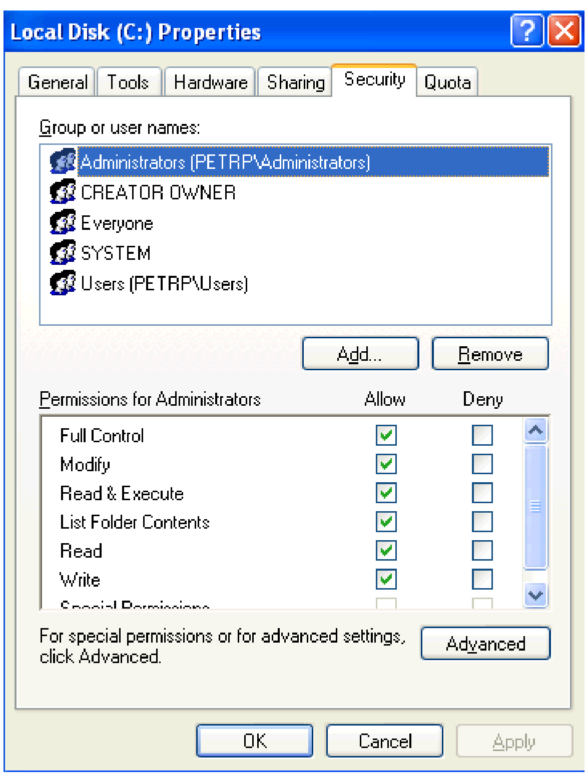
2. In **Windows 2000 and 2003**, the user account is the local account **NT Authority\Network Service** by default.

3. If you're using **Visual Studio 2005** 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.

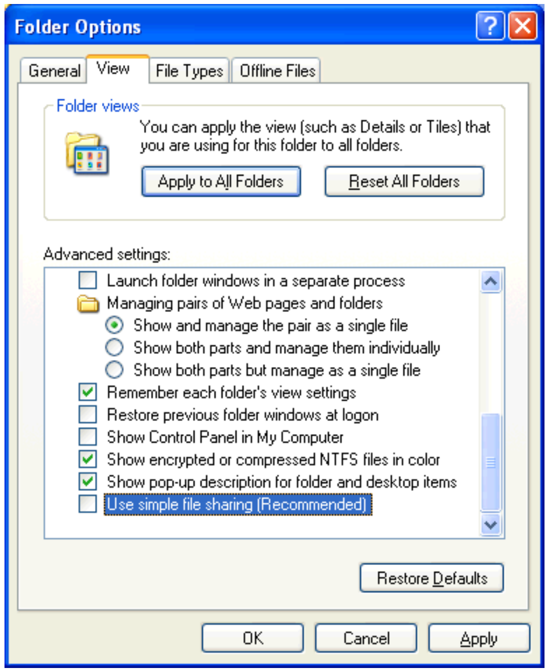
### Granting user account with Modify permission on Windows XP

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

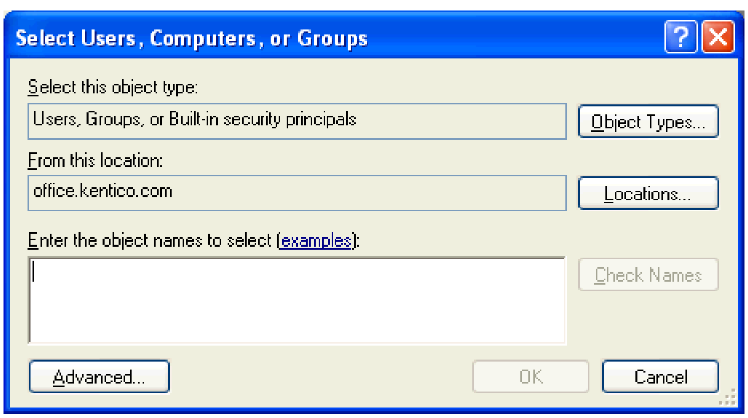


**Missing Security tab in folder properties dialog**

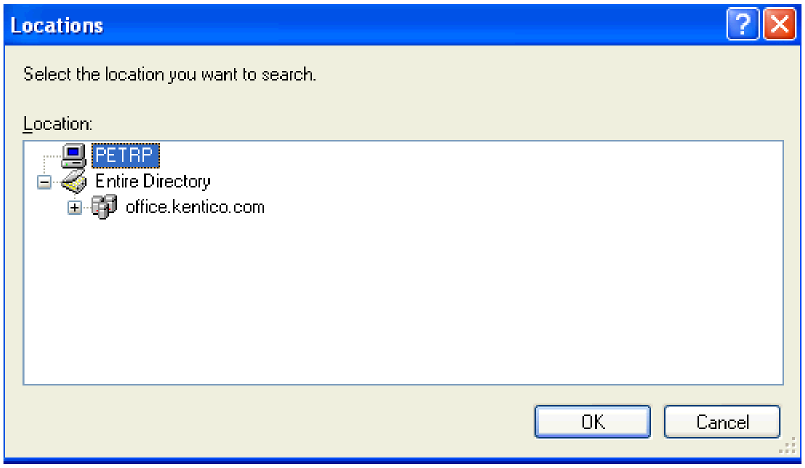
If you cannot see the Security tab, click **Tools -> Folder options** in the Windows Explorer main menu, choose the **View** tab and uncheck the **Use simple file sharing** box. Click OK. Now you should find the Security tab in the folder properties dialog.



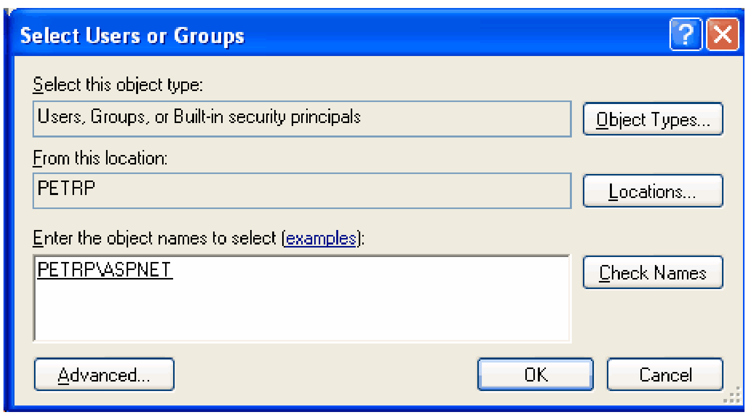

Click **Add...** The **Select Users, Computers and Groups** dialog appears.



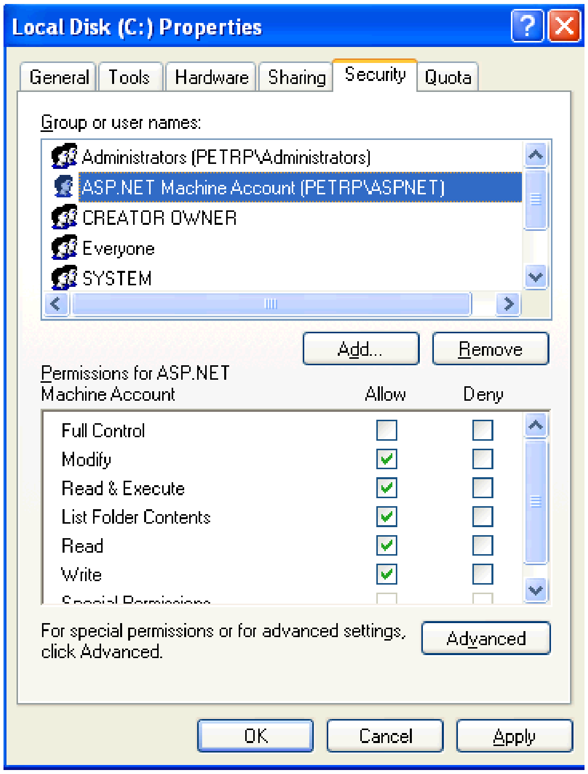
Click **Locations...** and choose your local computer:




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

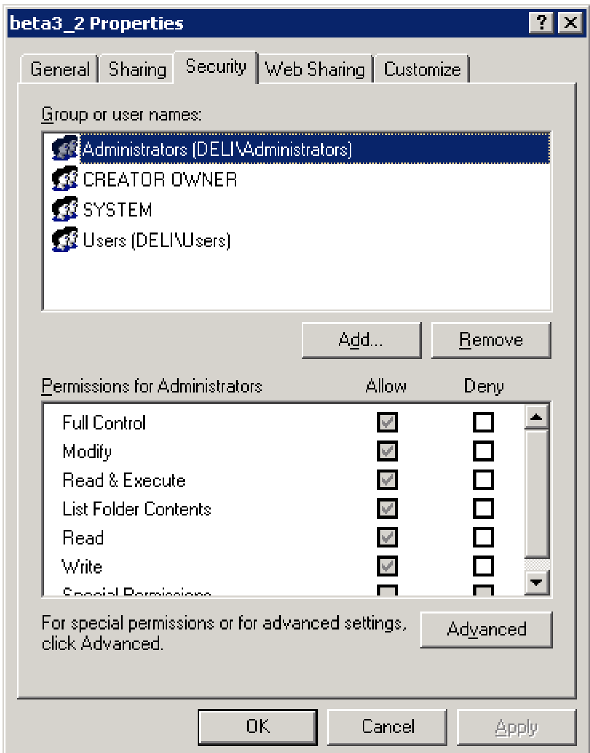


Click **OK**. The account is added to the list of accounts. Grant the account with **Modify** permissions and click **OK**.

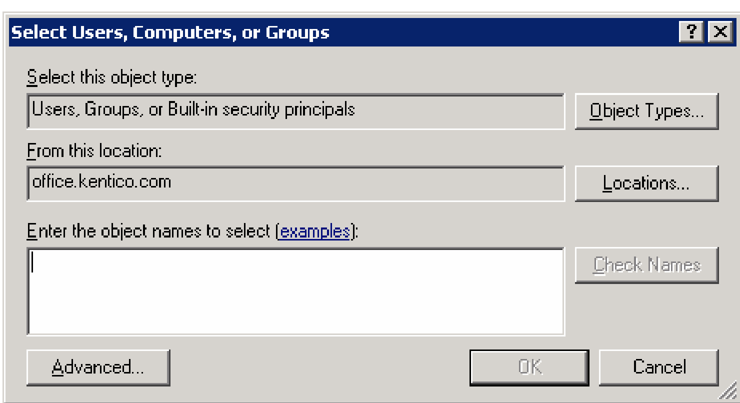


### Granting user account with Modify permission on Windows 2000/2003

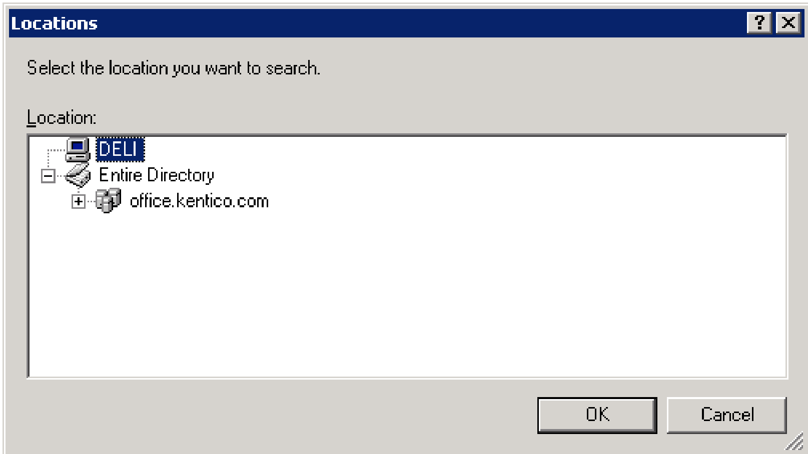
Open Windows Explorer, locate the folder with your web site, right-click the folder and display its **Properties**. Choose the **Security** tab.



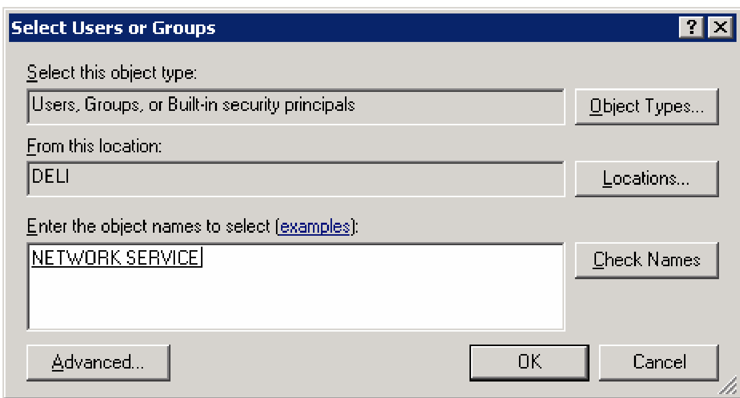

Click **Add...** The **Select Users, Computers and Groups** dialog appears.



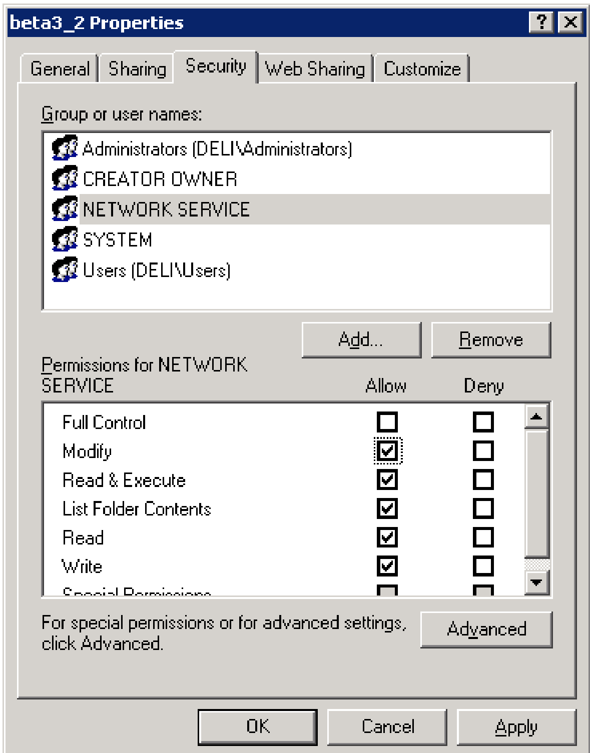
Click **Locations...** and choose your local computer:




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

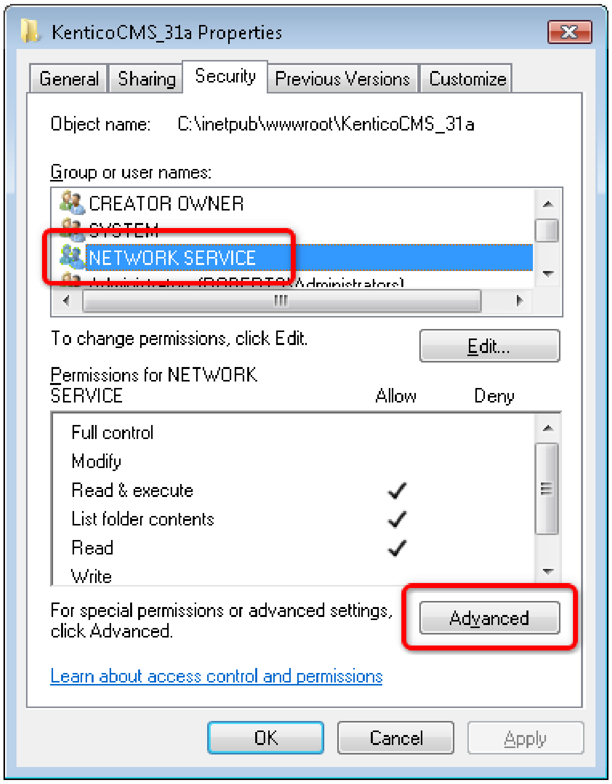


Click **OK**. The account is added to the list of accounts. Grant the account with **Modify** permissions and click **OK**.

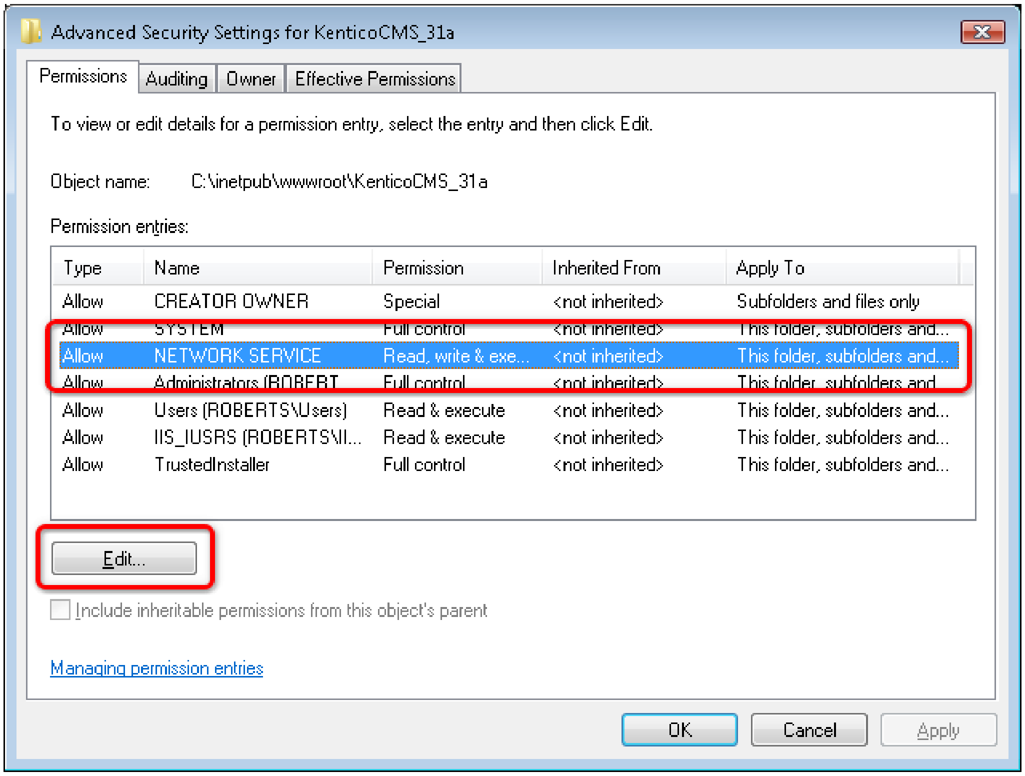


### Granting user account with Modify permission on Windows Vista/Server 2008

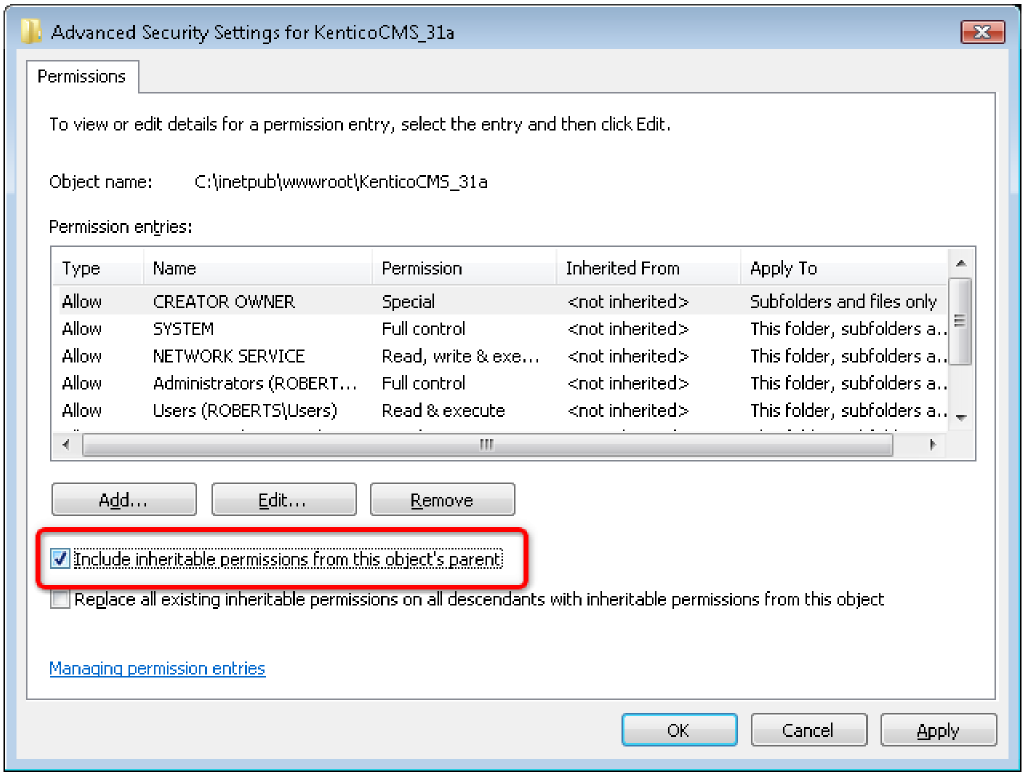
1) Open Windows Explorer, locate the folder with your web site, right-click the folder and display its **Properties**. Choose the **Security** tab. Select **NETWORK SERVICE** and click **Advanced**.



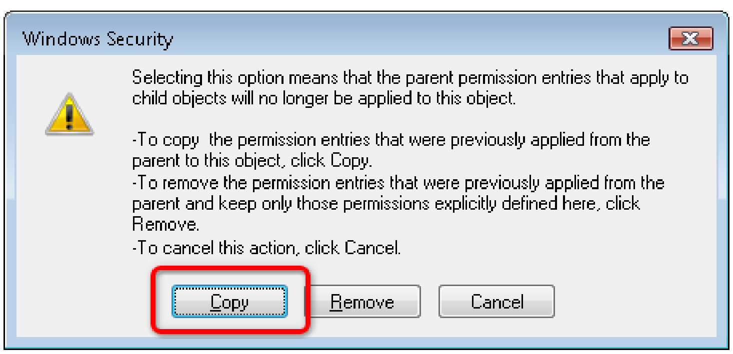
2) Select **NETWORK SERVICES** and click **Edit**.



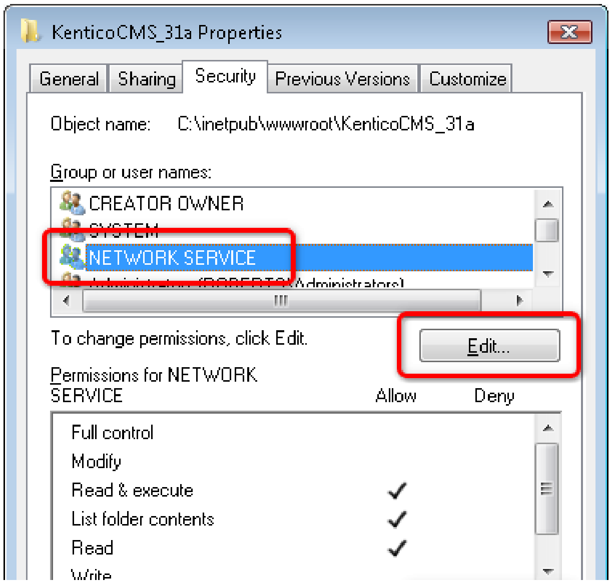
3) Disable the **Include inheritable permissions from this object's parent**.



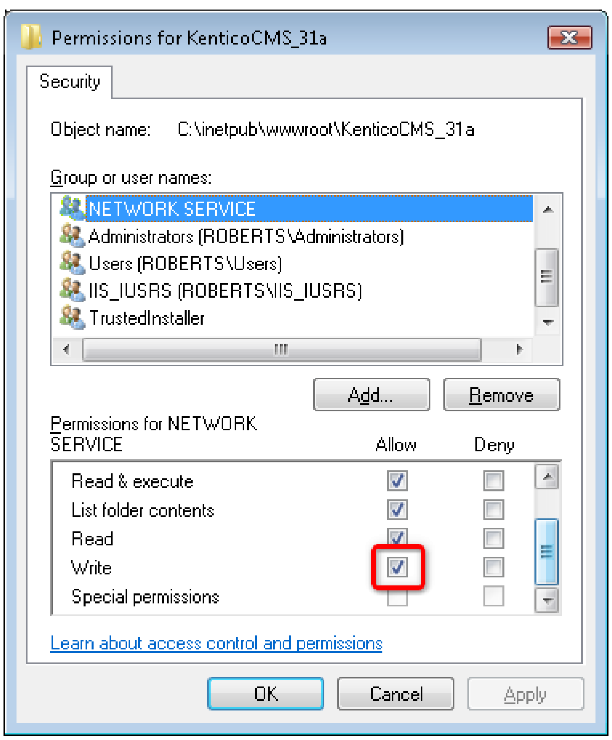

4) On the pop-up dialog, click **Copy**.



5) 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 **NETWORK SERVICES** and click **Edit**.




6) Check the **Allow** check-box for the **Write** permission and click **OK**.



7) You have assigned the NETWORK SERVICES account with the write permission. Kentico CMS should now be able to perform all disk write operations and therefore work correctly.

### 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" />**

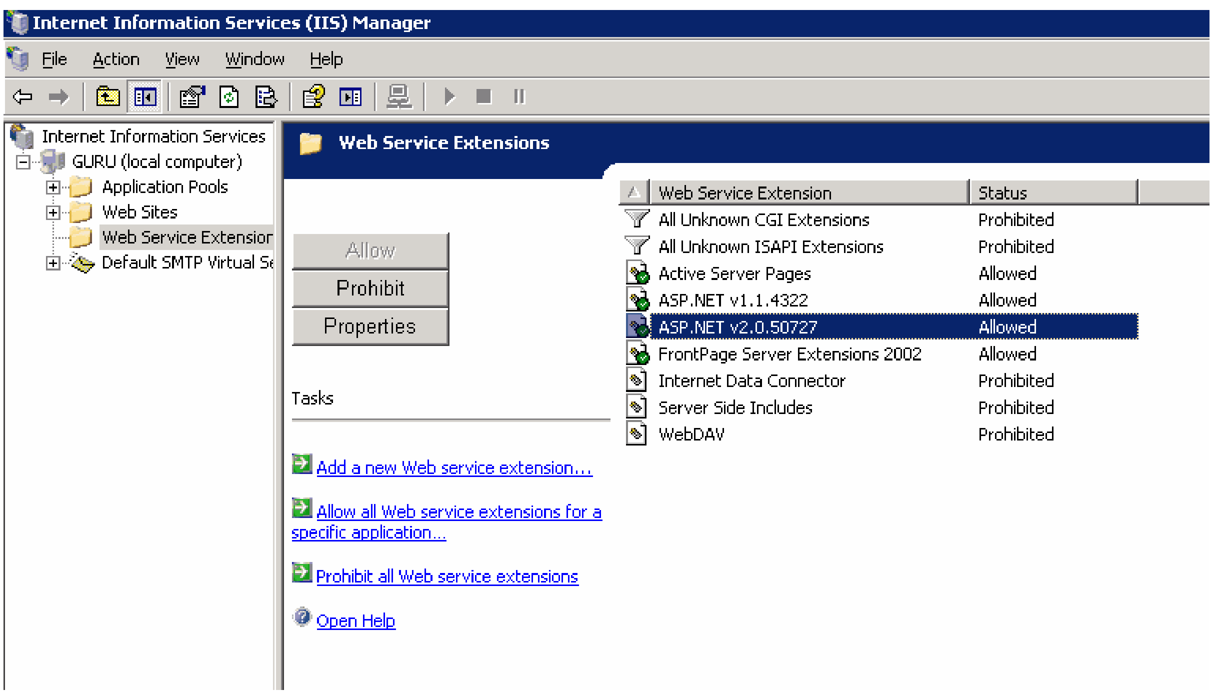
If you're running Kentico CMS on a shared hosting server, some providers require that you use the non-managed methods for directory operations:

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

## 3.15.4 ASP.NET not working on Windows Server 2003

If you get error Page not found or 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** as shown on the following figure:

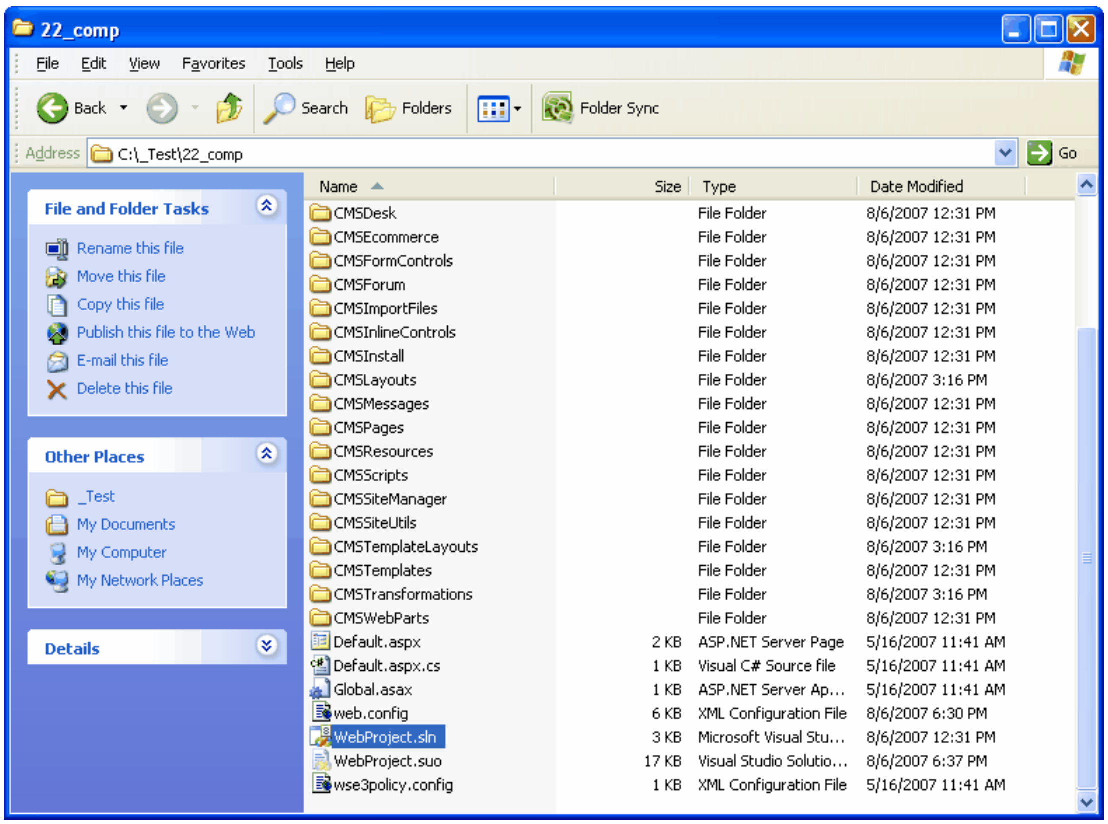


# Part IV

# 4 Visual Studio Integration

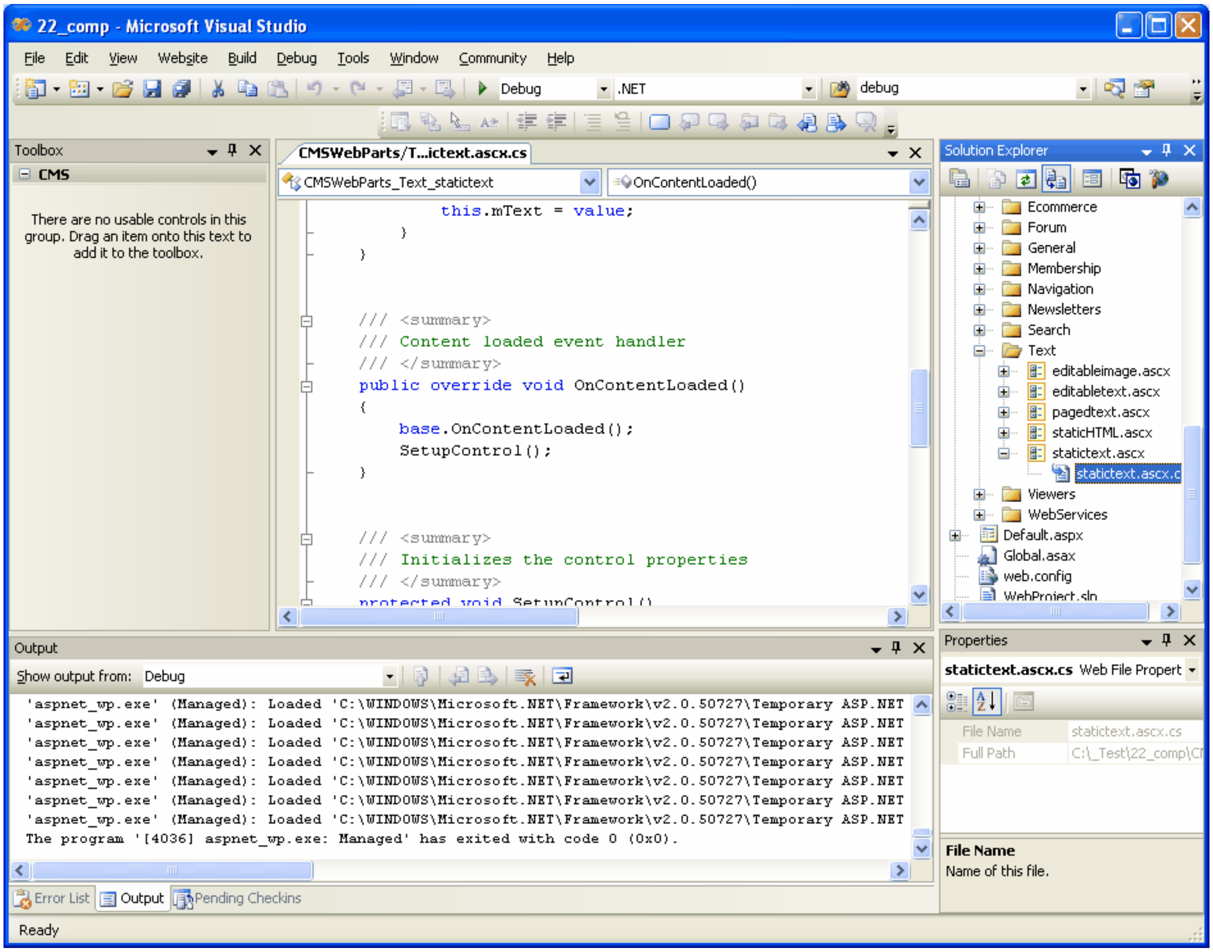
## 4.1 Opening the project

After you create a new Kentico CMS web project, you can open it in Visual Studio 2005 by double-clicking the WebProject.sln file in Windows Explorer:



If this option doesn't work, you can start **Visual Studio 2005** and choose **File -> Open -> Web Site** in the main menu and navigate to the folder that 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 web site**, 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 Unlimited License and Server License 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 keep other licensing conditions.

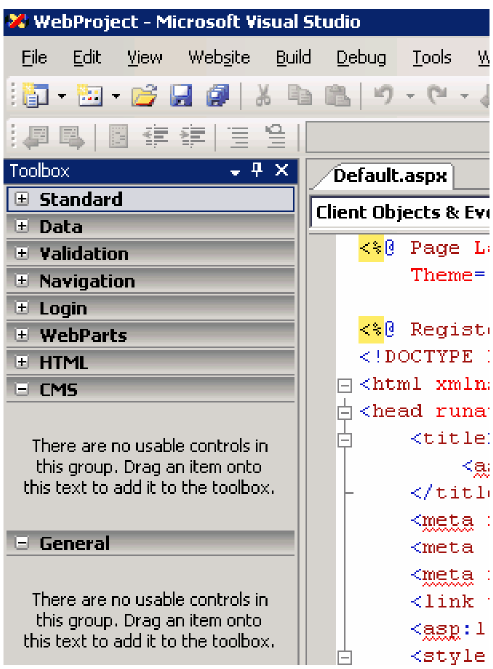
## 4.2 Adding Kentico CMS Controls to the Toolbox

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

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

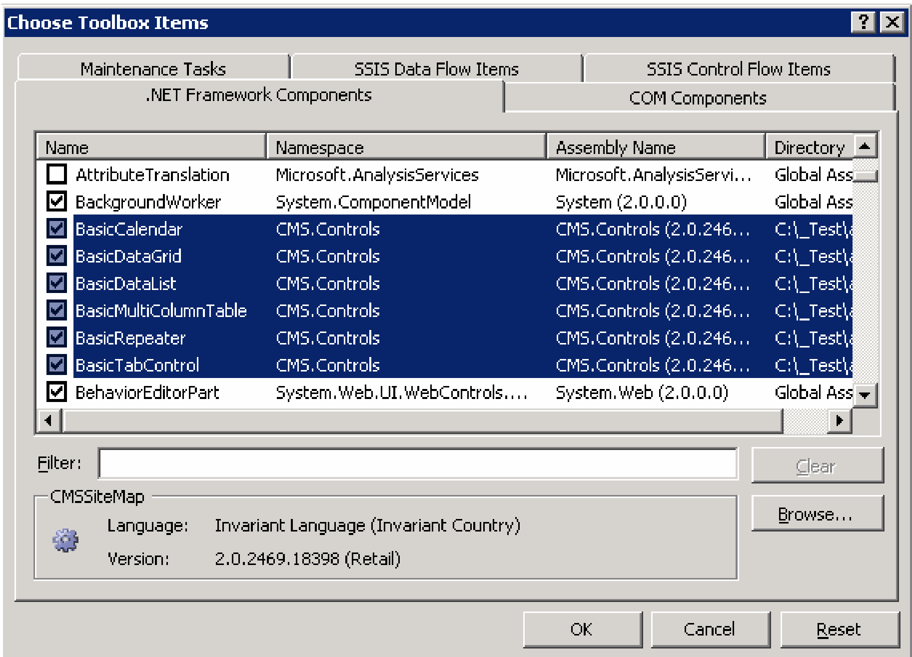
2. Right-click the **Toolbox** and choose **Add tab** from the context menu.

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

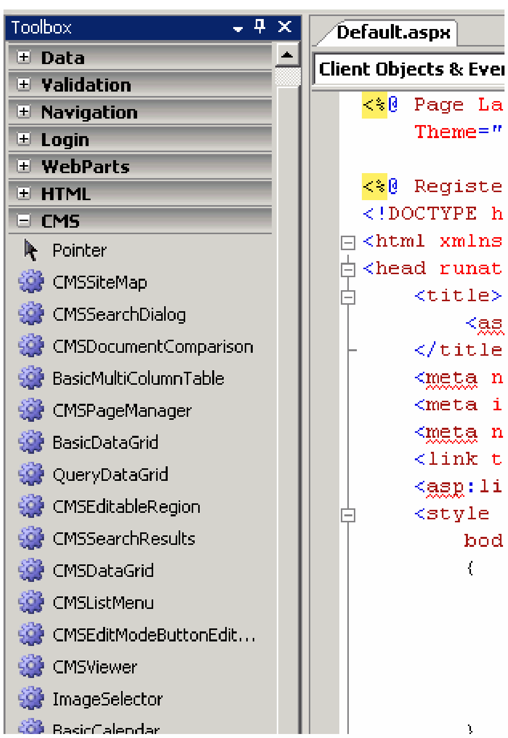



4. Right-click the new tab and choose **Choose items...** from the context menu.

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



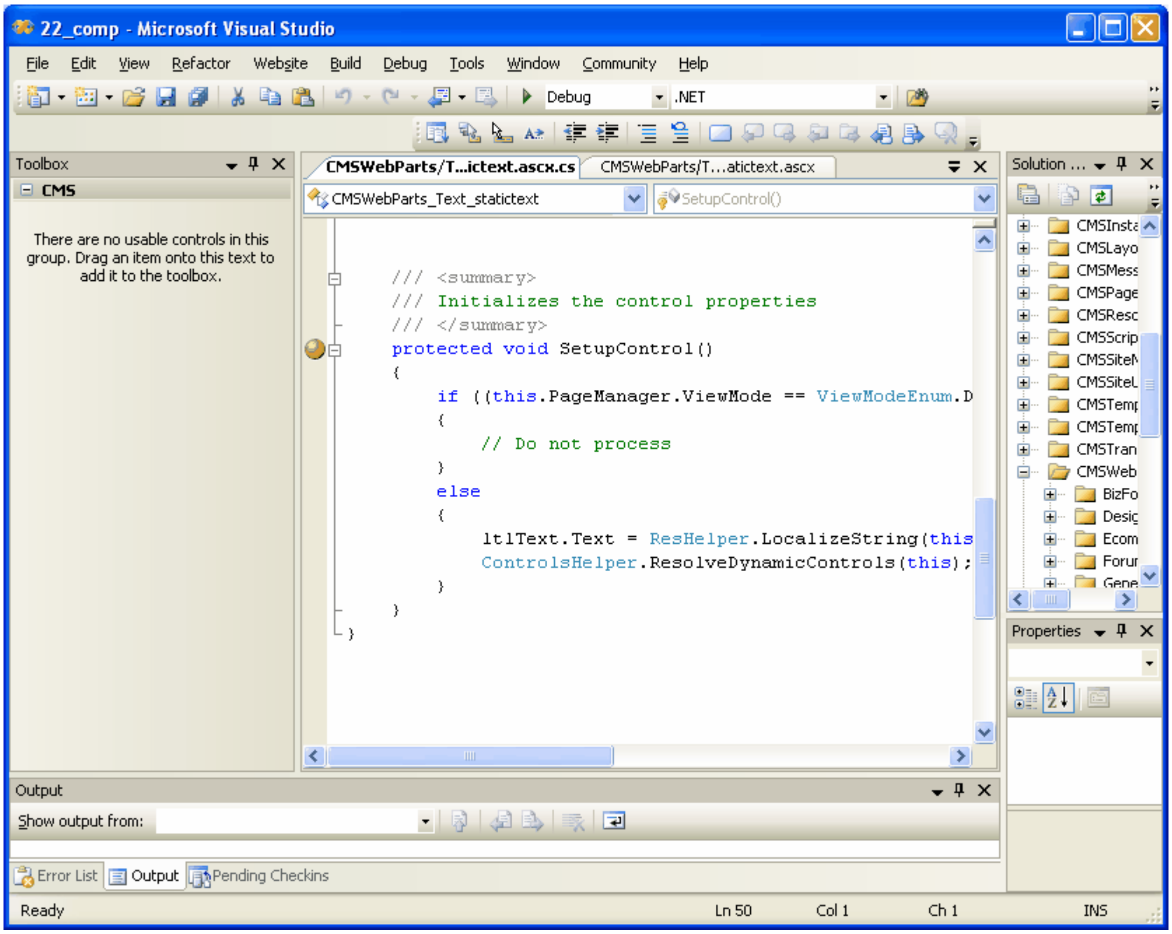

6. **The controls are now added to the Toolbox:**



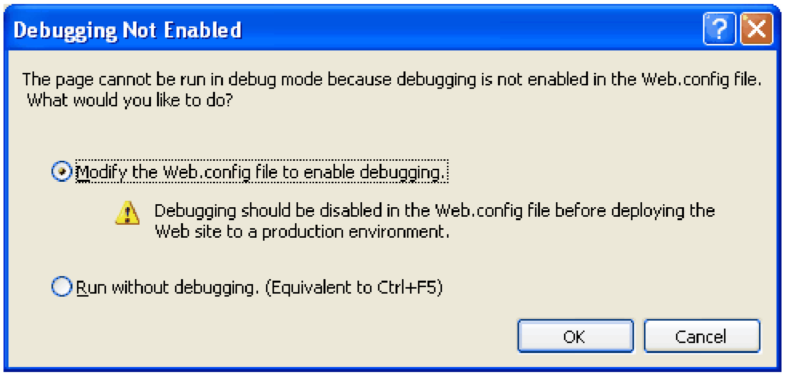

7. Now you can easily drag and drop the controls on your Web form.

# 4.3 Debugging

If you're adding custom code using Visual Studio 2005 into Kentico CMS, you can easily debug it in Visual Studio 2005 as you're used to since Kentico CMS is a standard VS 2005 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 web site starts and you can track the code flow. You may get a message like this:



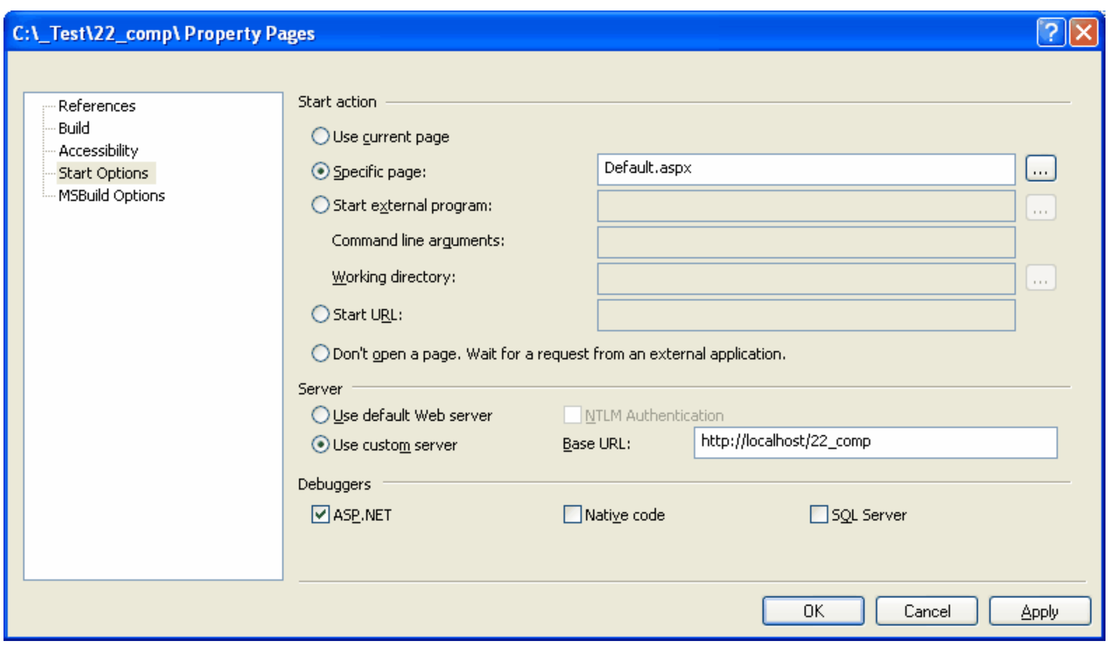

You will need to choose to **Modify the web.config file to enable debugging** and click **OK**. It's recommended that you disable debugging before deploying the web site to a production environment by setting the value

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

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

## Debugging using IIS web site

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




Please note that you need to be local administration and the web site or virtual directory must be configured for both anonymous and Windows authentication (see Creating a virtual directory for details).

# 4.4 Pre-compilation (Publish function)

If you want to pre-compile the web site before placing it on the live server, you can use the Publish function of Visual Studio 2005. 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 web site code, he 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 web site, you will not be able to use some of the Kentico CMS features:

- Import of the web site.
- 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 web site, 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.

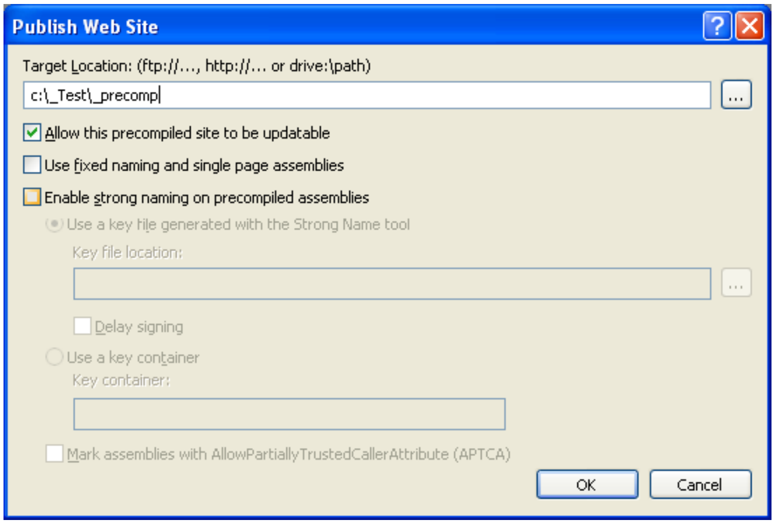
## Pre-compiling the Web Site

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

Since the virtual provider cannot run in a pre-compiled web site, 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 2005** main menu and pre-compile the web site. 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 web site 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 web site, 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 web site cannot be used for SQL Server database installation. Other options are:

- to install a non-compiled web site 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 web site 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 web site 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 web site before compilation**

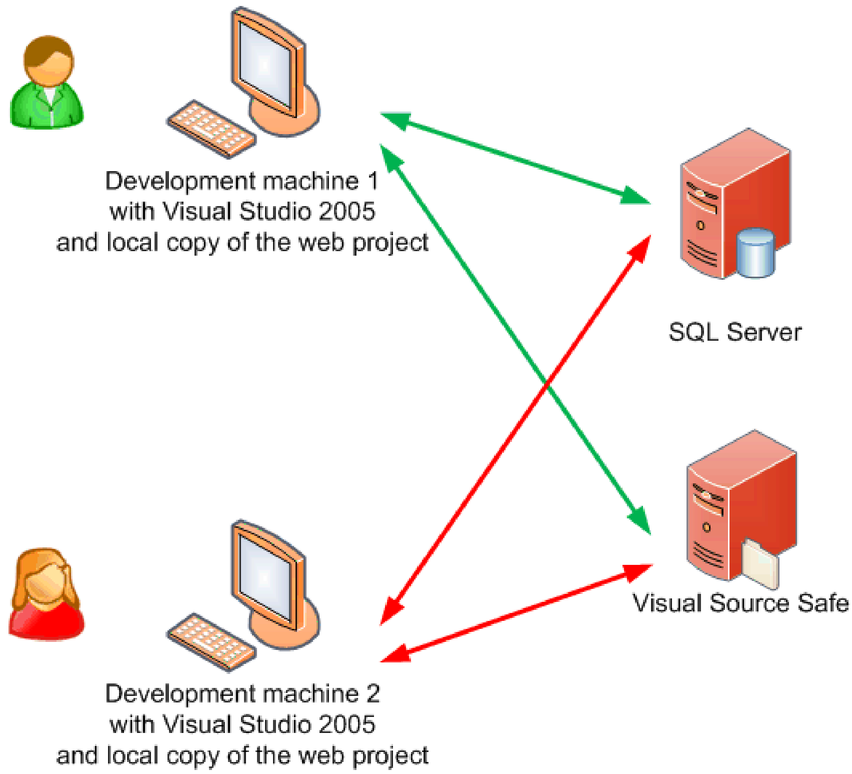
If you wish to check if your web site runs without using **VirtualPathProvider** (to simulate the precompiled environment), you can disable the provider by adding the following key into the appSettings section of your web.config file:
<add key="CMSUseVirtualPathProvider" value="false" />
If your system runs with VirtualPathProvider disabled, you should be able to run it after the precompilation or deployment process.

Related topics: Configuration for Medium Trust environment, Deployment to the live server, Installation on shared hosting server

## 4.5 Visual Source Safe and Team Development

Kentico CMS can be used in team development environment as any other web site project in Visual Studio 2005. You can also use Microsoft Visual Source Safe (VSS) as you're used to: you simply add the web site 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 2005

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 web site through the browser-based interface.

# 4.6 Opening VS2005 project in VS2008

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

- Start **Visual Studio 2008**
- Click **File -> Open web site...**
- Choose the folder with your web project on the disk and click **OK**.
- If you're asked if you wish to upgrade the web site to .NET Framework 3.5, choose **Yes**.

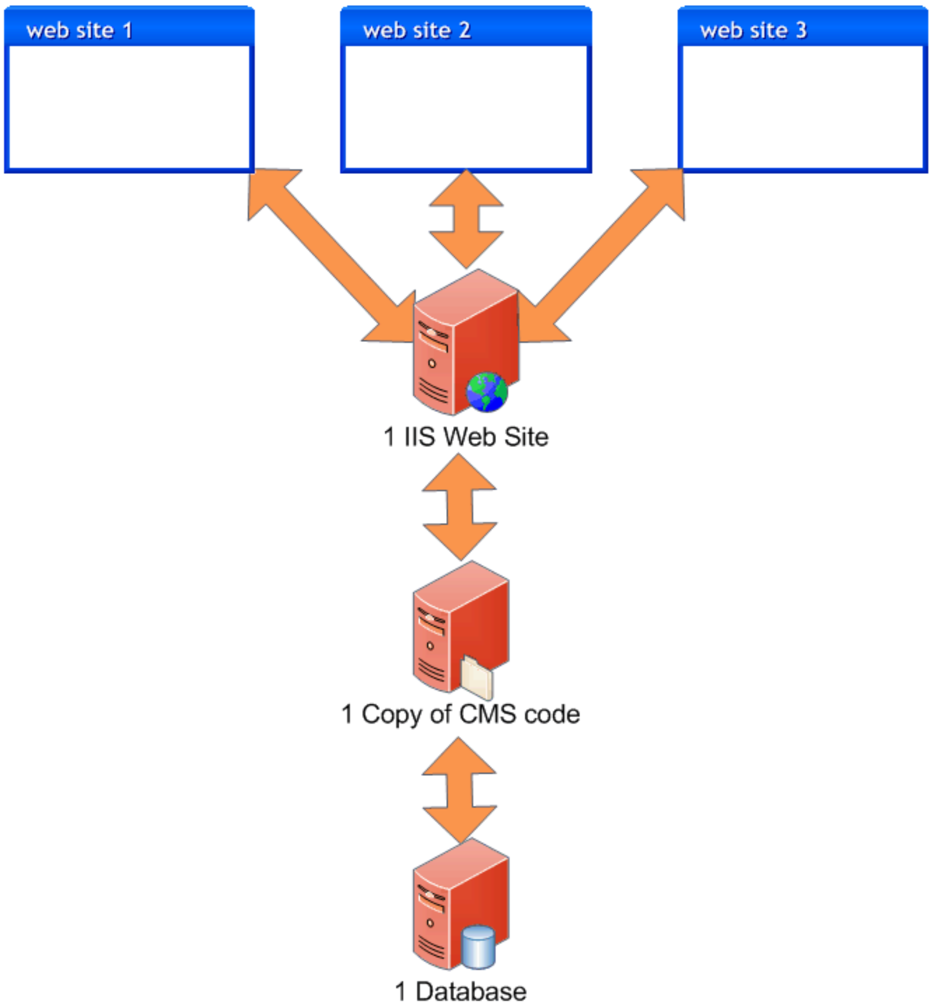
You may receive a compilation error saying there are different versions of the **System.Web.Extensions.dll** library in the Global Assembly Cache (GAC) and a temporary folder. In this case, you need to locate the file **bin/System.Web.Extensions.dll** in your web project and delete it.

# Part V

# 5 Managing sites and settings

## 5.1 Site Management Overview

Kentico CMS allows you to manage multiple sites in a single installation. The database contains data for all web sites and the web sites 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 web sites.




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 web sites 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 web site:

- You build web sites 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 web sites by administrator's mistake.

## 5.2 Managing sites

You can manage the web sites in **Site Manager -> Sites** dialog.
***Creating a new web site***
Please see New site wizard for more details on creating a new web site.

**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 web site. Use domain name without http:// protocol and without www. If you use other port than 80, specify it as well.<br><br>Correct:<br>mycompany.com<br>partners.mycompany.com<br>mycompany.com:8080<br><br>Incorrect:<br>http://mycompany.com<br>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 the visitor if she doesn'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 web site for internal use. |

**Domain aliases tab**
Here you can add domain aliases that point to this web site.

**Example of domain aliases**

If your web site uses **mycompany.com** domain as the primary domain and you also use **my-company.co.uk** domain name that points to the same web site, you need to add **my-company.co.uk** alias to the list of domain aliases.

**Culture tab**

Here you can choose in which languages the web site content is available. You will use this tab to configure web site that provides content in multiple languages.

## 5.3 Starting and stopping sites

You can run and stop web sites 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 web sites on a single domain**, such as http://localhost, you need to specify the domain (localhost) for multiple web sites 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 web sites in chapter Configuring multiple web sites.

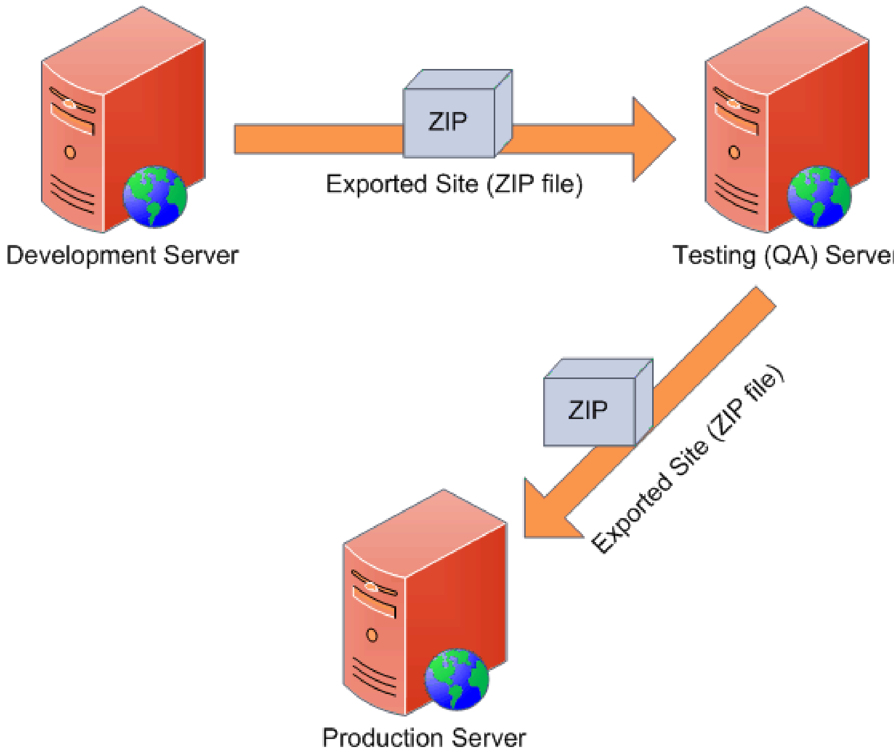
## 5.4 Creating a new site

Please see New site wizard.

# 5.5 Export and Import

## 5.5.1 Export and Import Overview

You can export and import web site content and settings from one Kentico CMS instance to another. You can use this feature to move web site or chosen objects between development, testing (QA) and production (live) server as illustrated in the figure below:

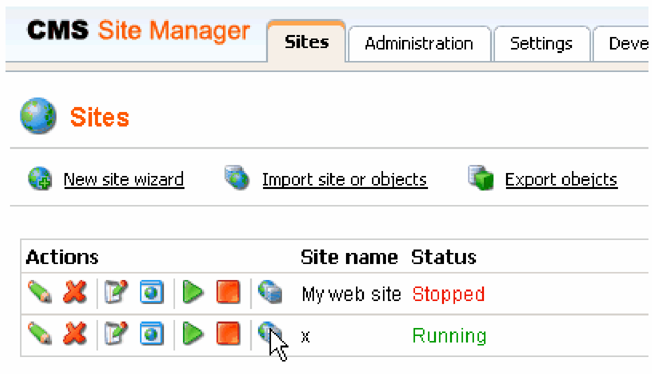


## 5.5.2 Exporting a site

Kentico CMS allows you to export whole web site 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.

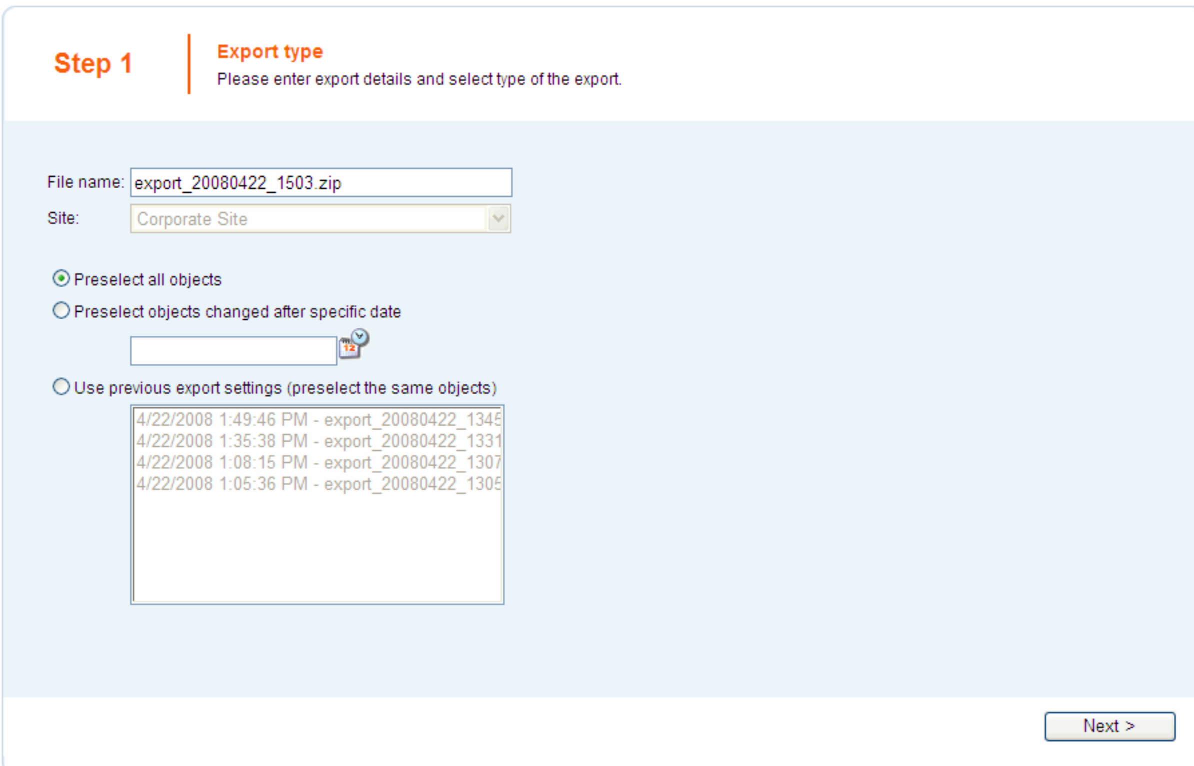


In the first step of the wizard, you have to fill in the name of the export package and choose type of objects pre-selection:

**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.



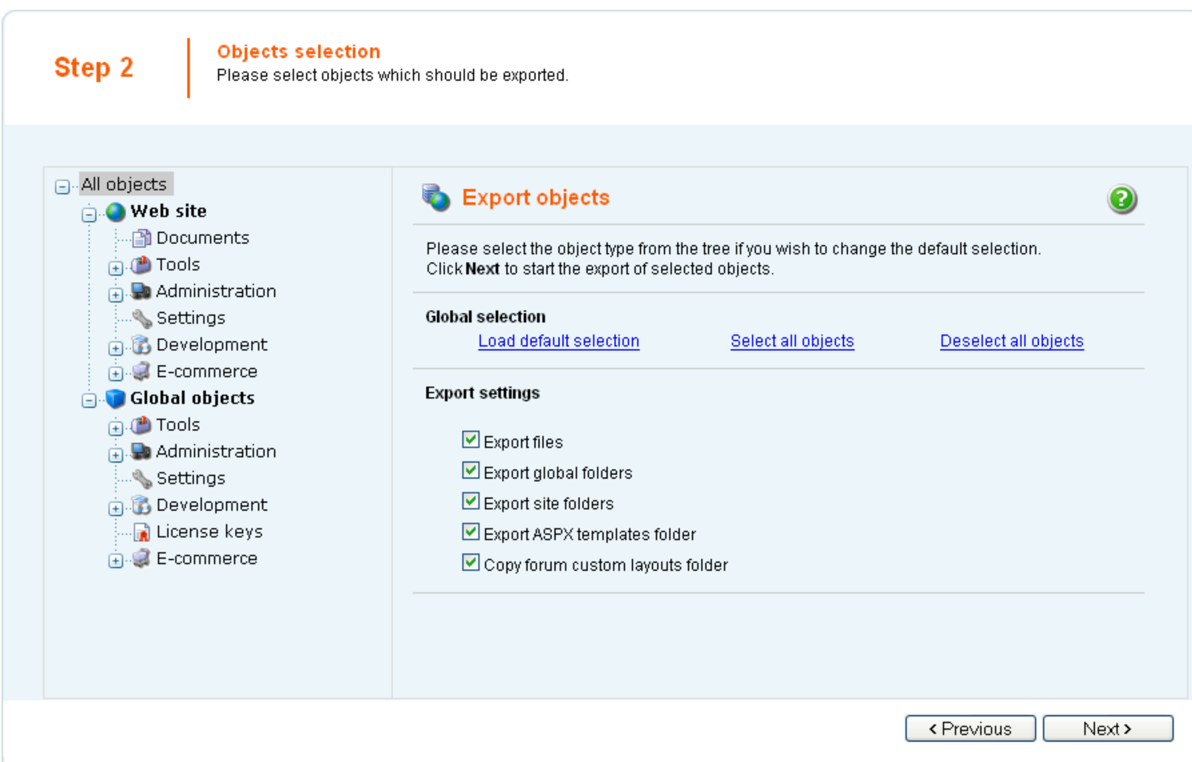
In **Step 2**, you can select which objects will be exported. The tree on the left represents object categories. These reflect the sections of the administration interface under that the objects can be found. The **Site** category contains objects related to the selected web site. The **Global objects** 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*
- **Copy forum custom layouts folder -** if checked, folder with custom forum layouts will be exported
  - *<web project>\CMSModules\Forums\Controls\Layouts\Custom*



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

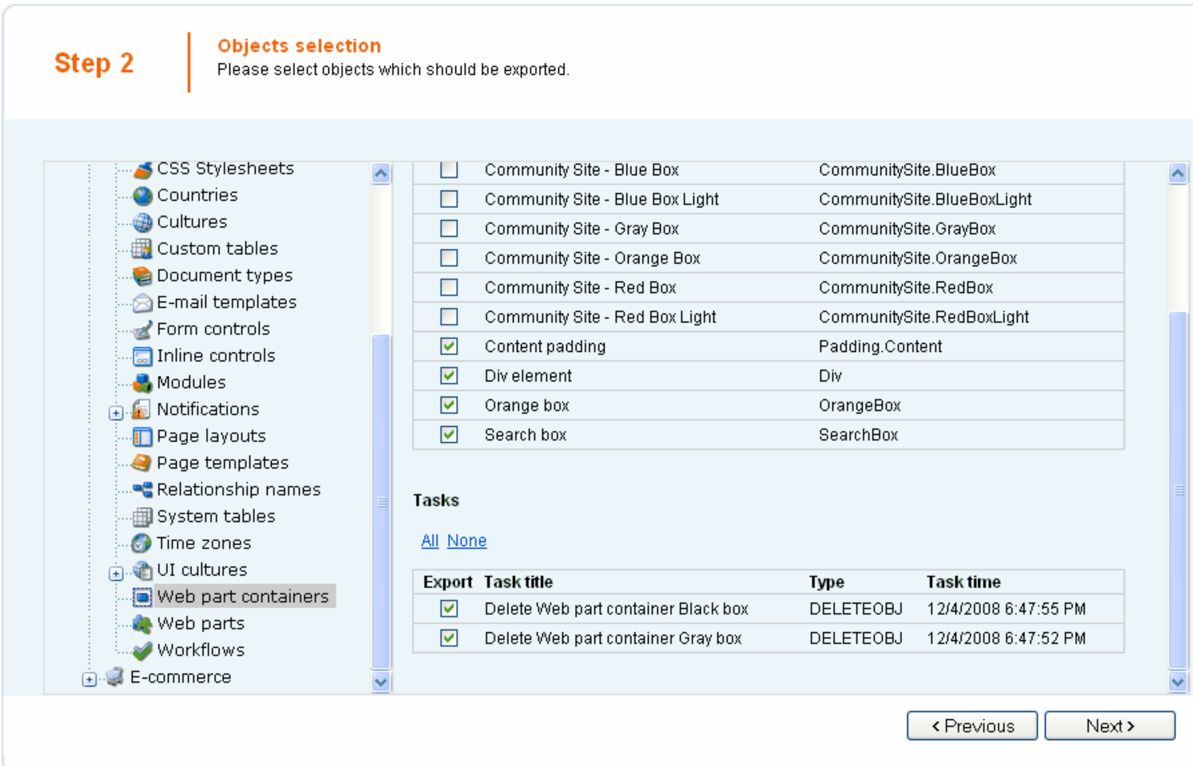
### Message boards

- **Export board messages** - if checked, board messages will be exported together with particular message boards

### Media libraries

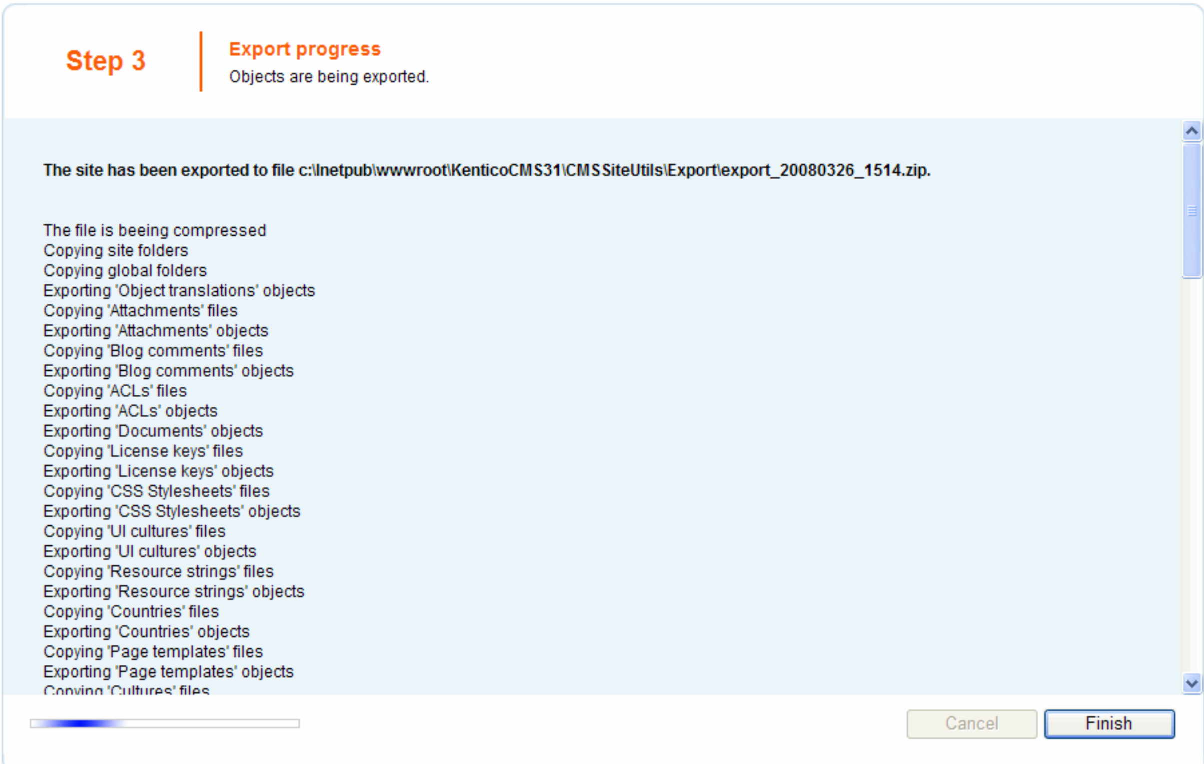
- **Export media files** - if checked, media files stored in the database will be exported
- **Export physical files** - if checked, physical media files stored in the file system will be exported; this option is not selected by default as it may cause the package size grow extremely large; instead, 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**.



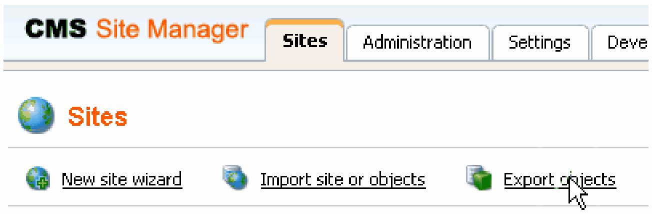

Now you can import the site on some other machine using the **Import site wizard** described in chapter Importing a site or objects.

## 5.5.3 Exporting objects

Besides exporting the whole web site, 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.

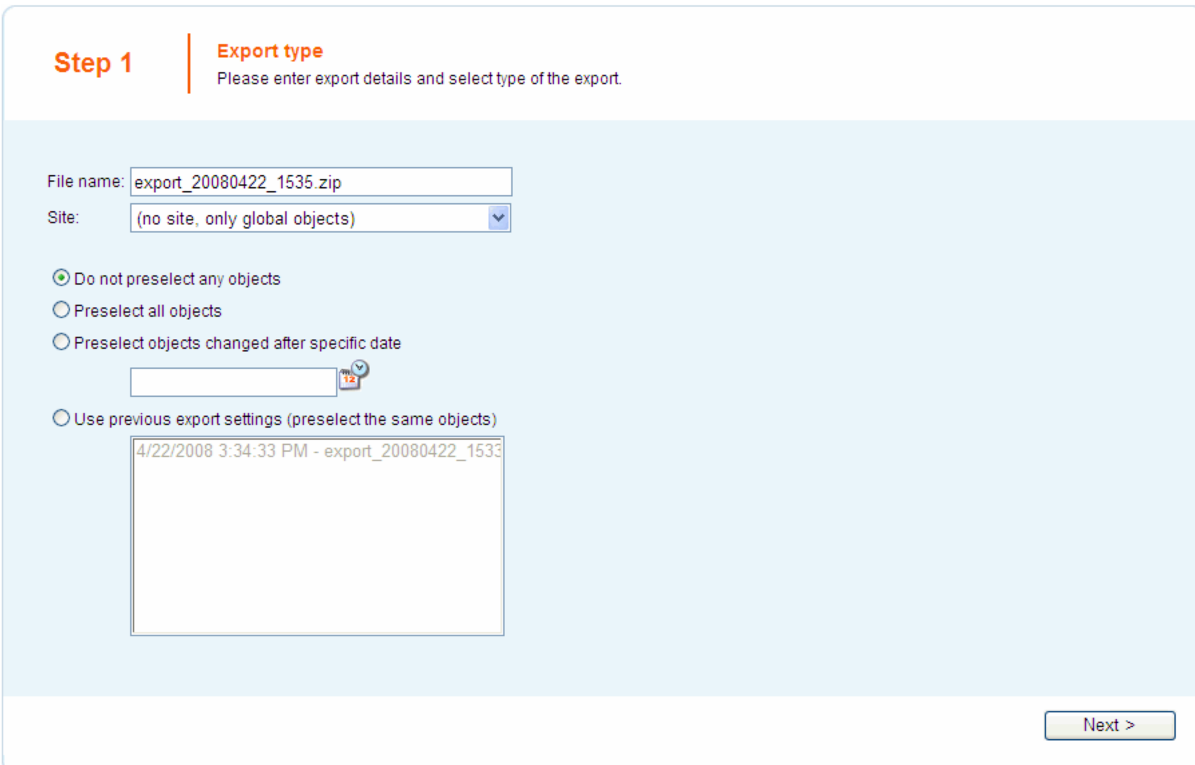


In the first step of the wizard, you have to fill in the name of the export package and choose type of objects pre-selection:

**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.



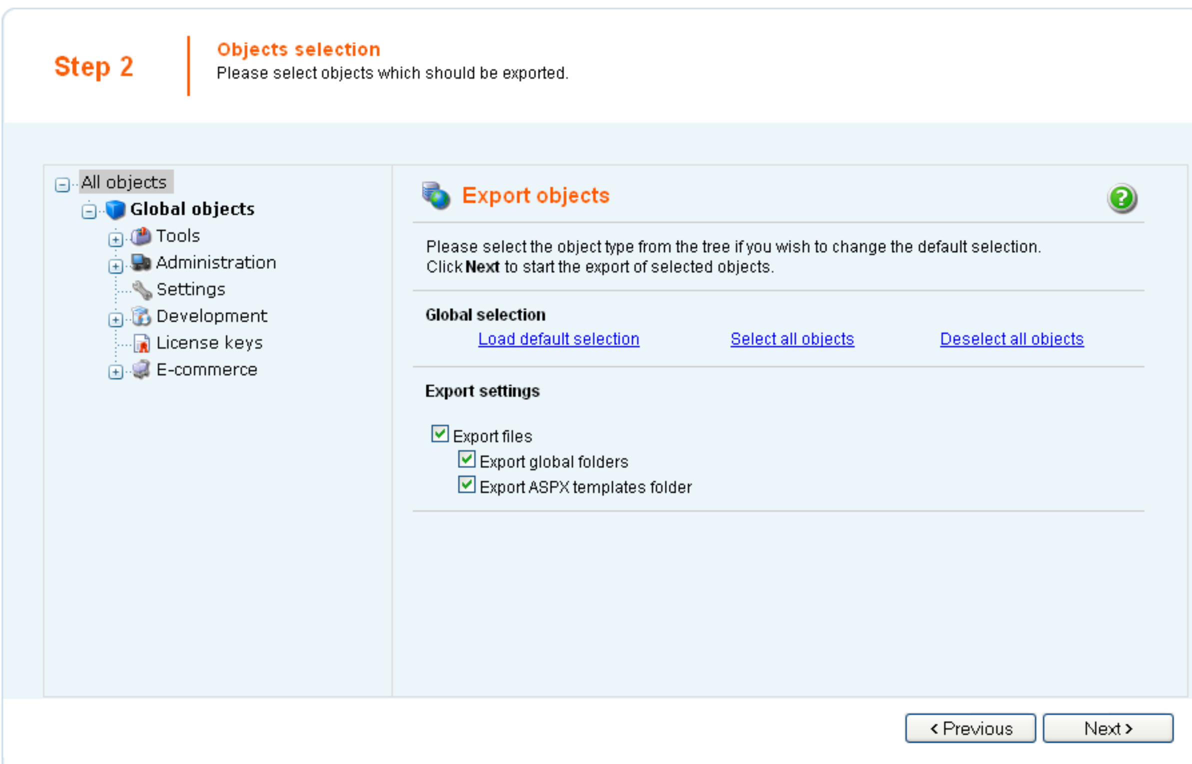
In **Step 2**, you can select which objects will be exported. The tree on the left represents object categories. These reflect the sections of the administration interface under that the objects can be found. The **Site** category contains objects related to the selected web site. The **Global objects** 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**

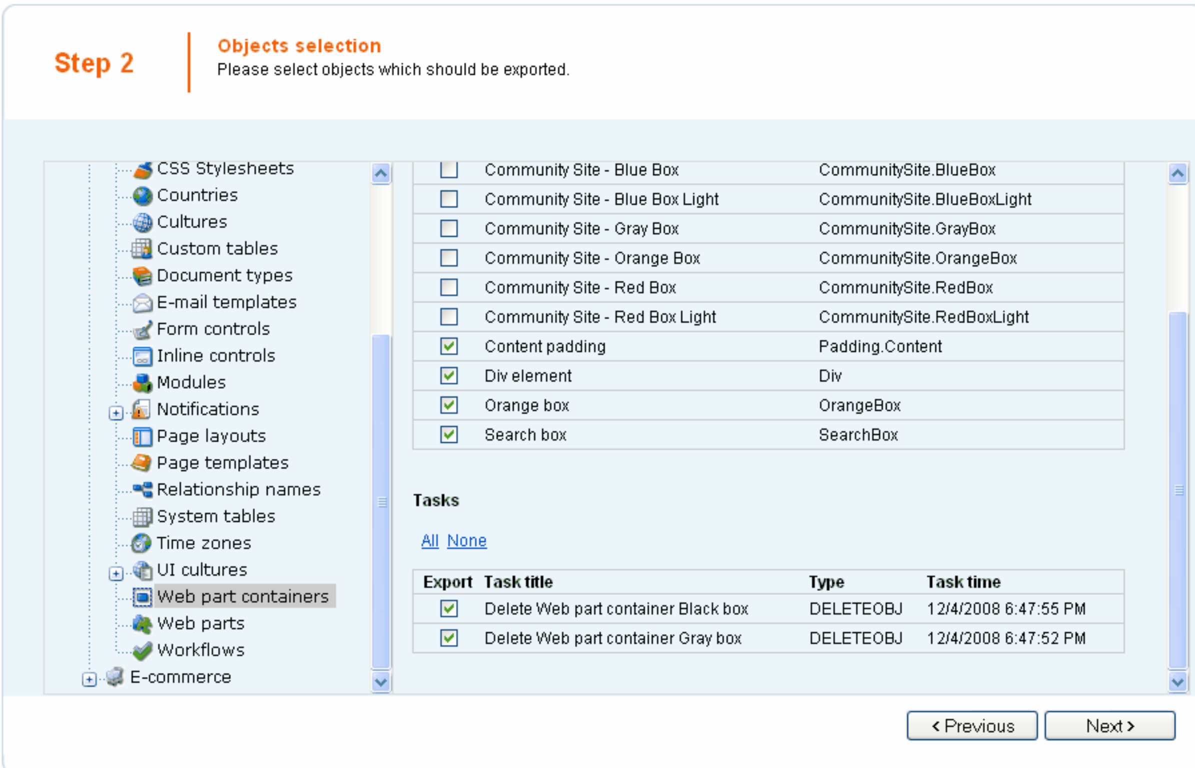
- **Load default selection** - objects will be selected based on the pre-selection choice made in Step 1
- **Select all objects** - all objects will be selected
- **Deselect all objects** - no objects will be selected

**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*

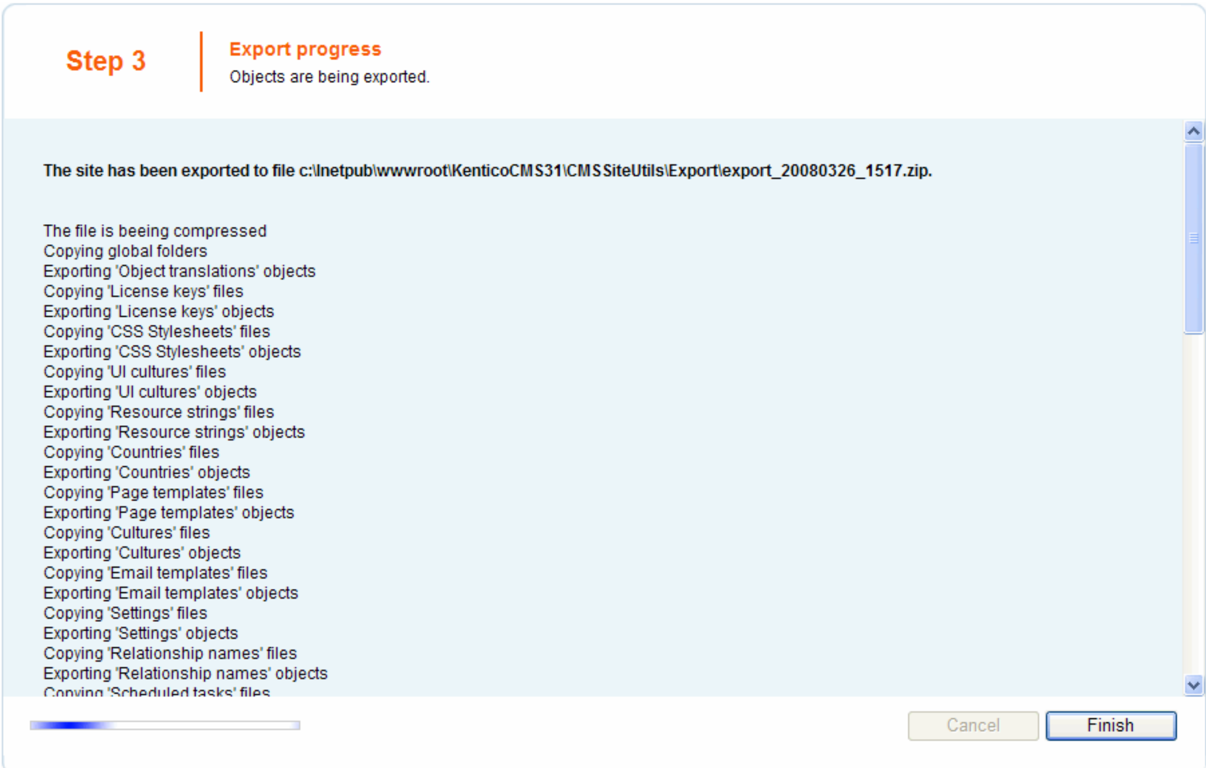


If you have the "**Log export tasks**" option enabled in **Site Manager -> Settings -> Content Staging**, a list of object deletion tasks may also be displayed at the bottom of the list. This happens when some objects have been deleted (just as the two Web part containers in the screenshot below). If 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 following chapter.

## 5.5.4 Exporting single objects

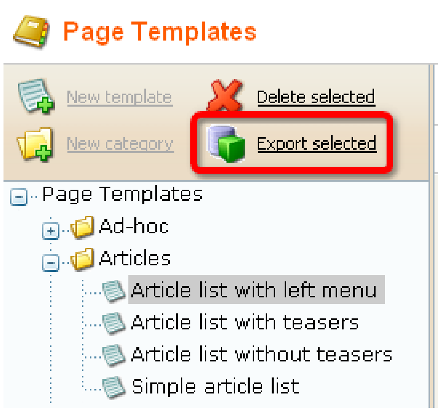
Some objects can also be exported separately as single object packages. This is useful when you want to quickly export only one or few objects apart from the rest of all other objects.

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.

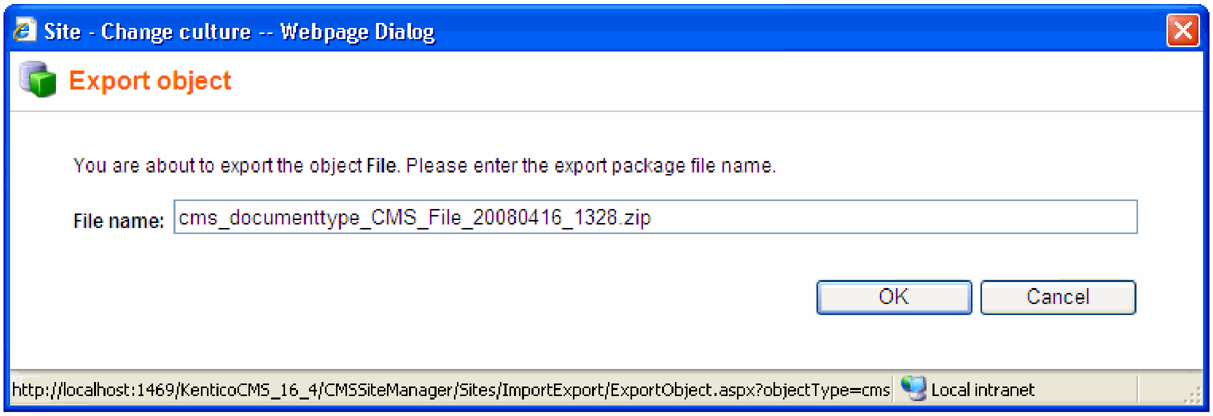
These objects can be found in the corresponding sections in **Site manager -> Development**. You can export an object by clicking the **Export object** ( ) icon.

| Actions | Display name ▲ | Code name |
|---|---|---|
| | Article | CMS.Article |
| | Blog | CMS.Blog |
| | Blog month | CMS.BlogMonth |
| | Blog post | CMS.BlogPost |
| | Cell phone | CMS.CellPhone |
| | Community - Transformations | Community.Transformations |

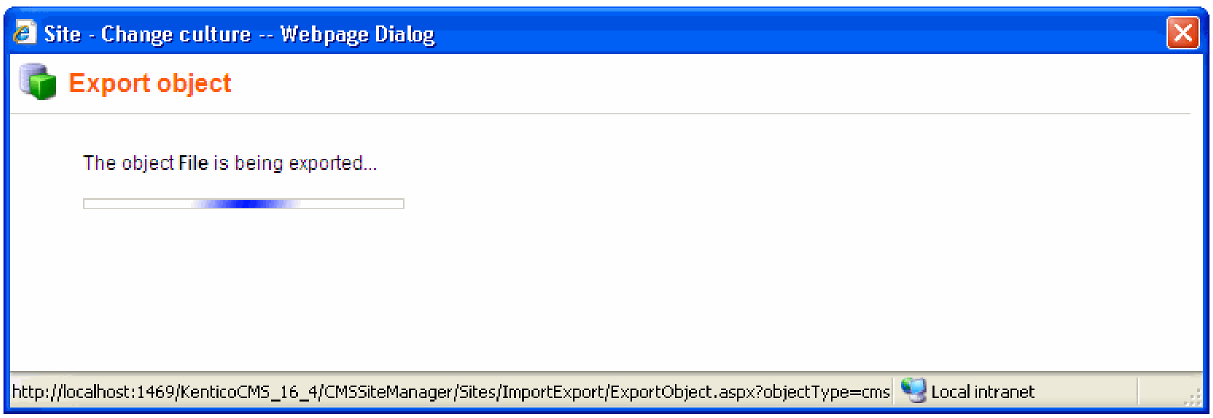
Or the **Export selected** link in case of Page templates and Web parts.



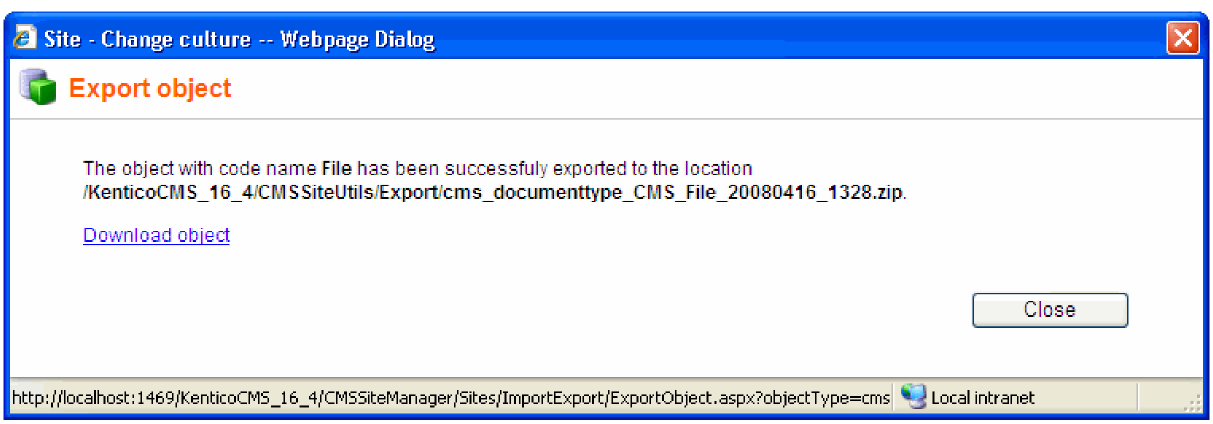
The following window will appear. Enter the name of the export file (default name will be pre-entered) and click **OK**.




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.



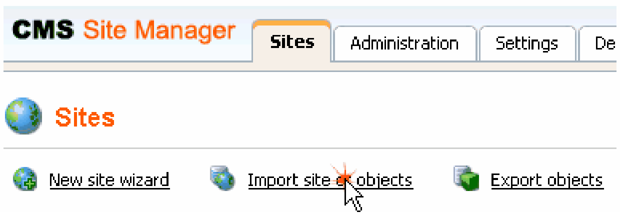

Finally, close the window by clicking the **Close** button.

## 5.5.5 Importing a site or objects

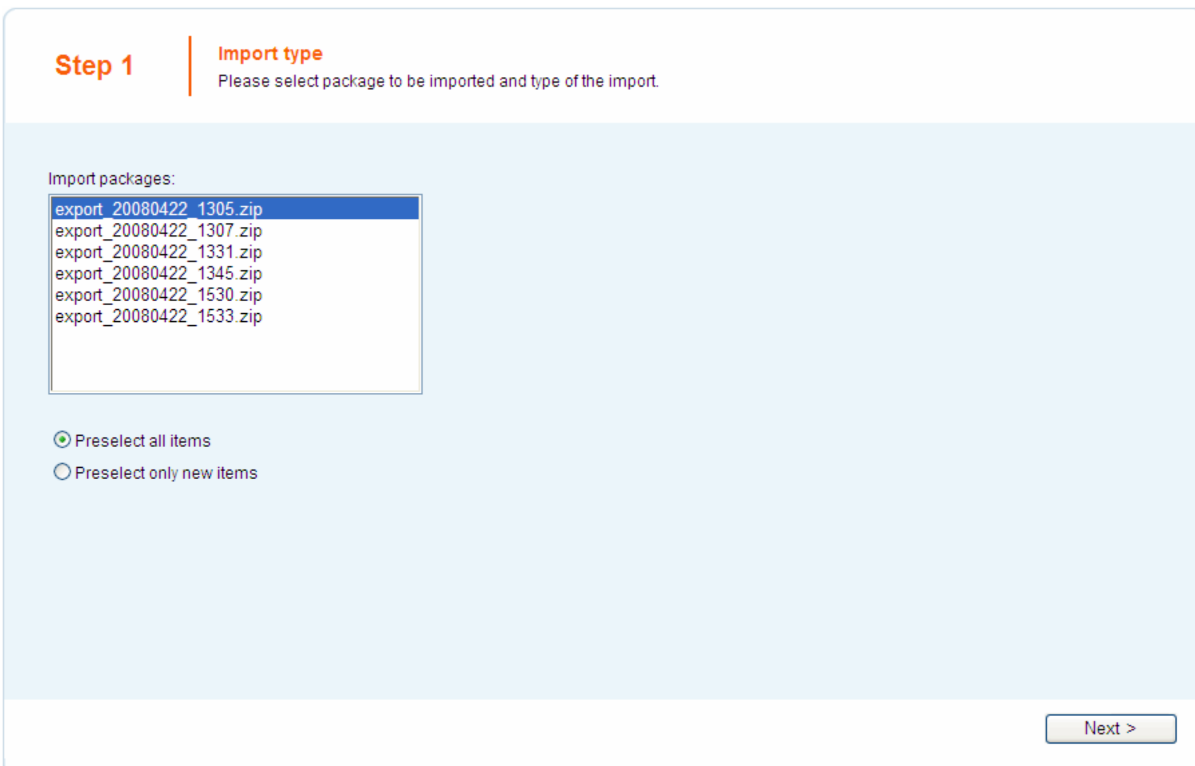
After you export some site or objects using the **Export site wizard** or **Export objects wizard**, you can import it using the **Import site or objects wizard**. Before you start the wizard, you need to copy your exported packages into the **<web project>\CMSSiteUtils\Import** folder.

### Importing a site

Go to **Site Manager -> Sites** and click **Import site or objects**.




Choose to import the appropriate package and click the **Next** button. Using the two radio buttons below, you can determine if duplicate items will be preselected or if only new items will be selected for import in the next step.

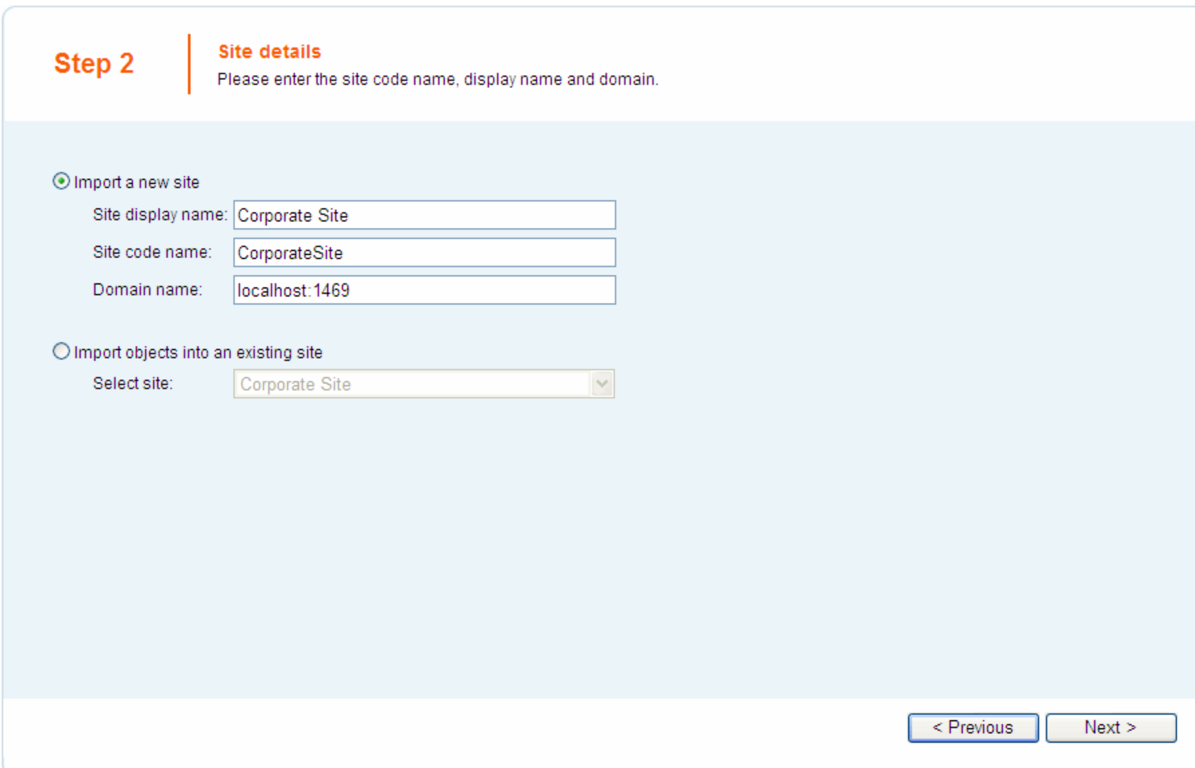


**Step 2** will be displayed only when you are importing a site package. In case that you are importing a package containing only global objects, Step 3 will be displayed instead.

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.

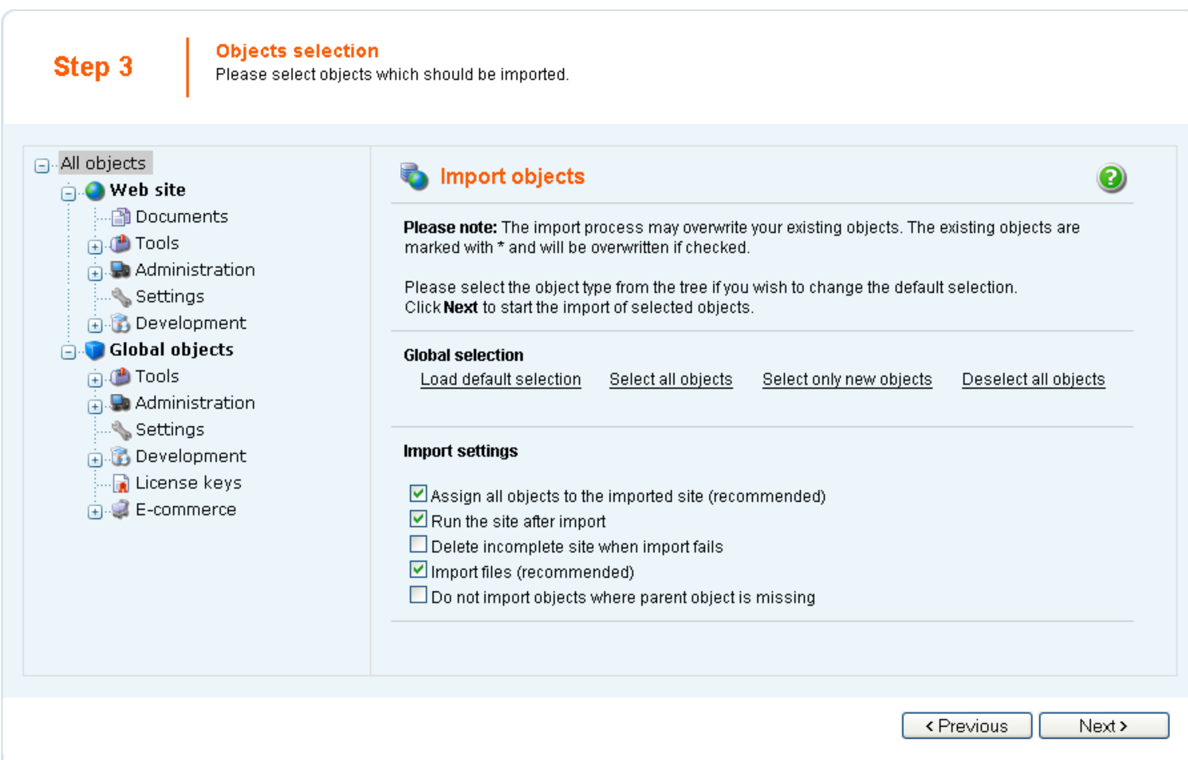


In **Step 3**, you can select which of the objects from the package will be imported. The tree on the left represents object categories. These reflect the sections of the administration interface under that the objects can be found. By selecting a category, a set of check boxes appears in the right part 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:

- **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
- **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

- **Overwrite system queries** - this check-box might be displayed optionally in case that you are importing packages from versions prior to 4.0; 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

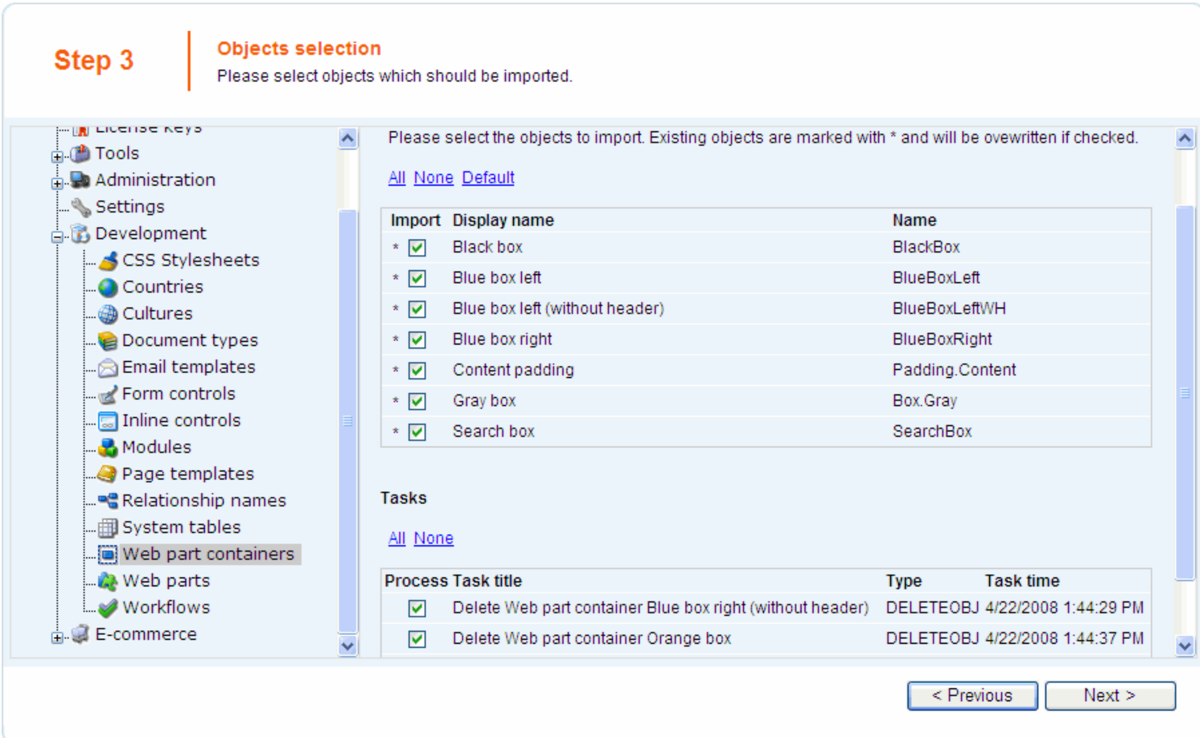
### Message boards

- **Import board messages** - if checked, board messages included in the package will be imported together with the message boards

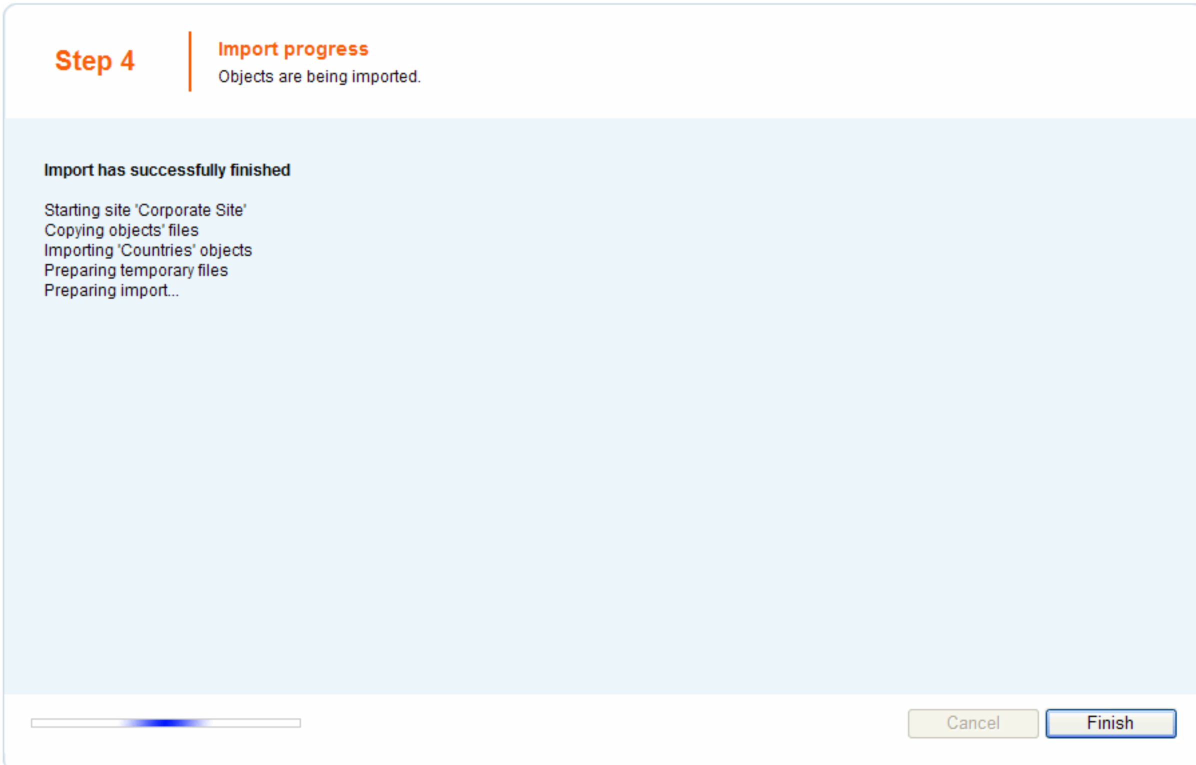
### Media libraries

- **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.

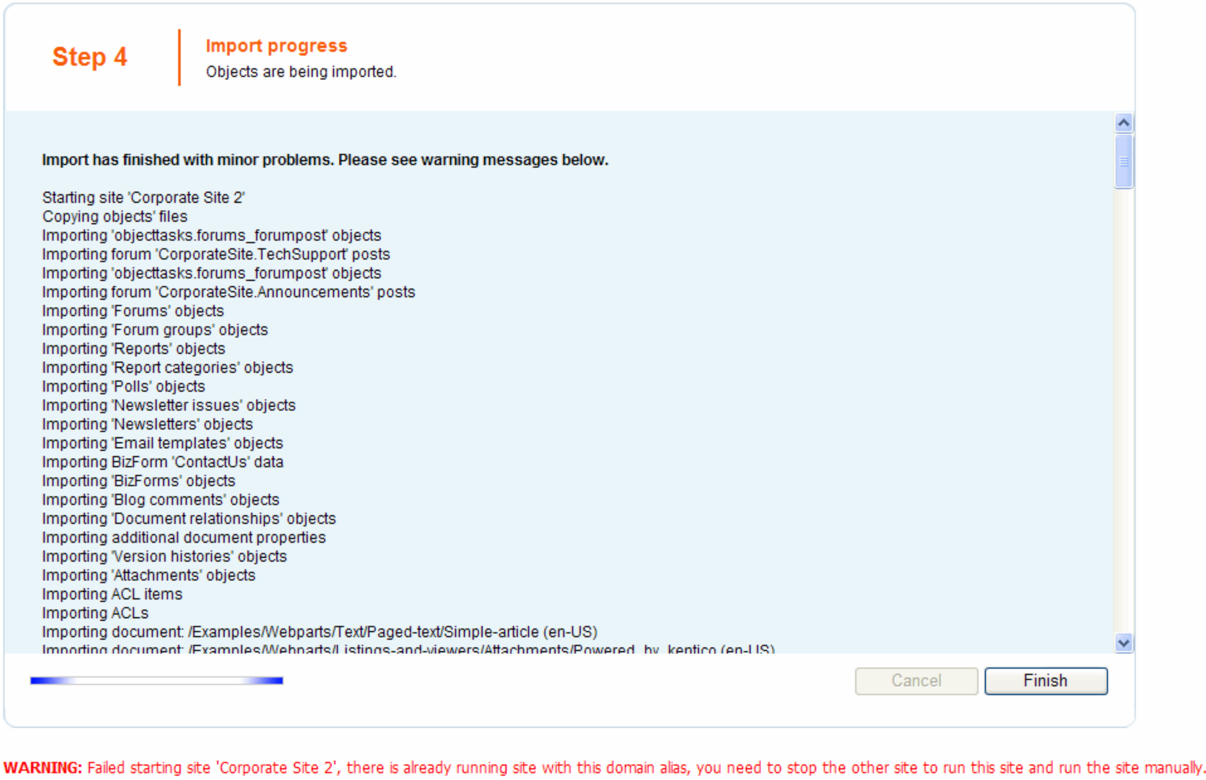



**Please note**

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 web sites 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 web site 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 web site 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 web sites in a single installation. The following figure describes the main folders and how they are affected during the import and export:

- **App_Browsers**
- **App_Code**
  - **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_Themes**
  - **<stylesheet name>** (all folders related to stylesheets assigned to or used on the web site)
- **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 web site, 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 web site, 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)
- **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)

## 5.5.7 Import/export troubleshooting

### 5.5.7.1 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**.

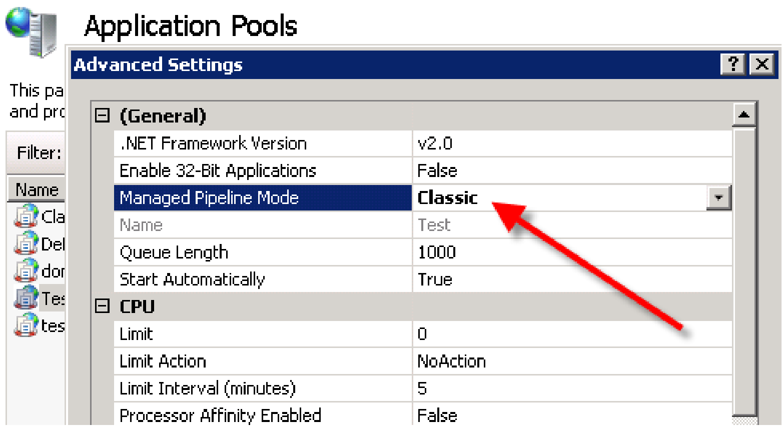
2. Open **Application pools** section.

3. Select application pool of your site and select **Advanced settings...** .

4. For property **Managed Pipeline Mode** select **Classic** instead of **Integrated**

or

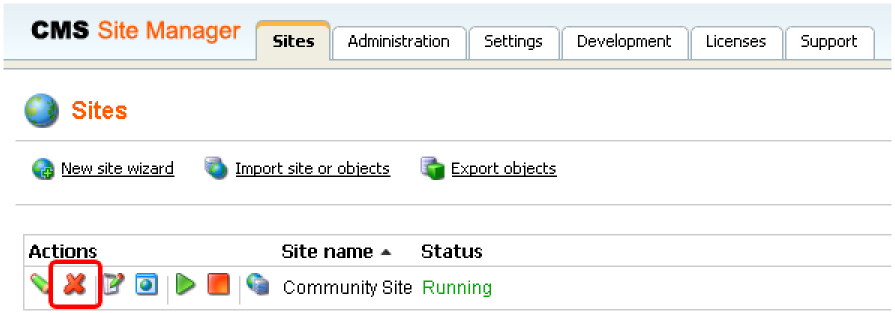
You can select **Classic .NET AppPool** for your web site.



## 5.6 Deleting sites

You can delete sites in the system in **Site Manager -> Sites**.

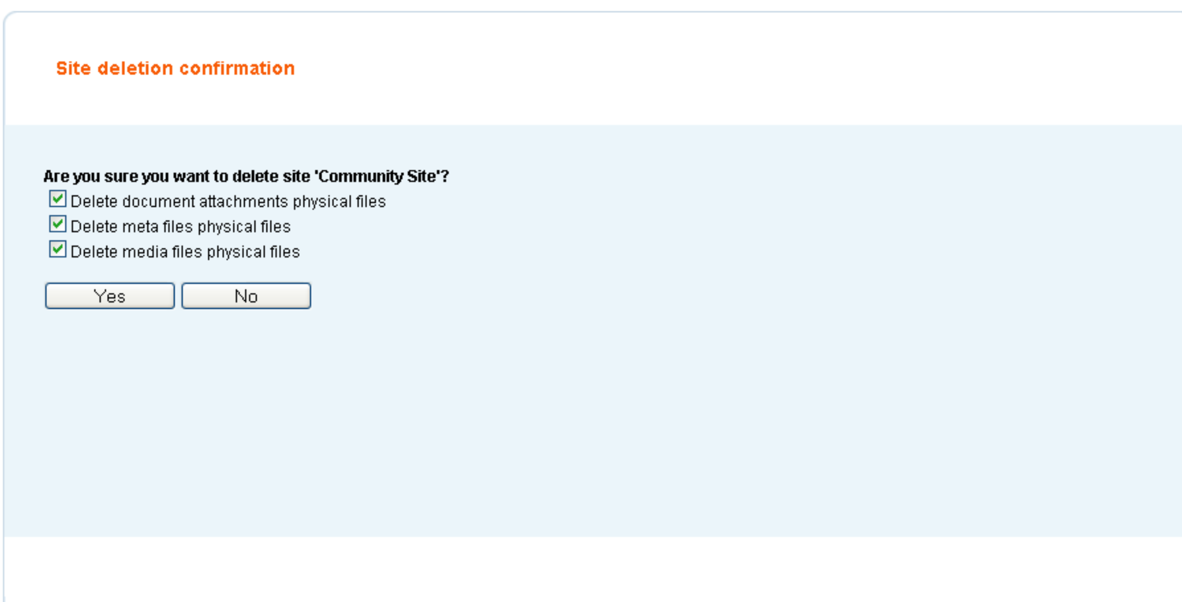
1) Clicking the **Delete** ( ) icon of the site that you want to delete.



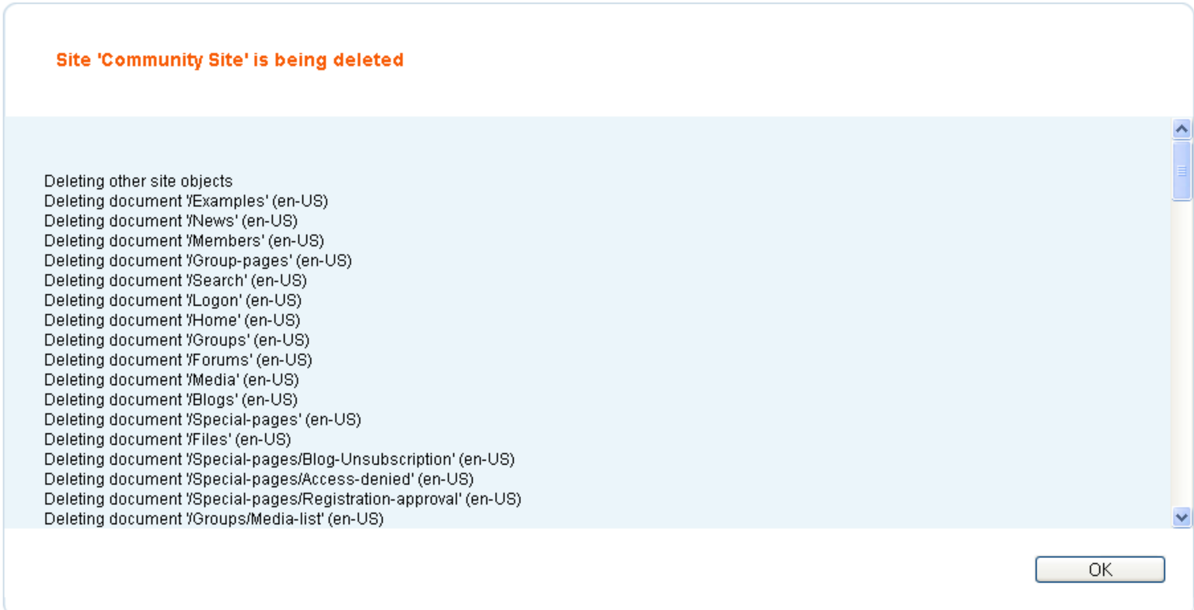

2) In the **Site deletion confirmation** dialog, you can select the following options:

- **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.

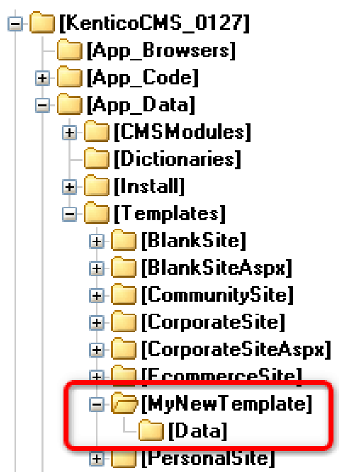


## 5.7 Creating web templates

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:

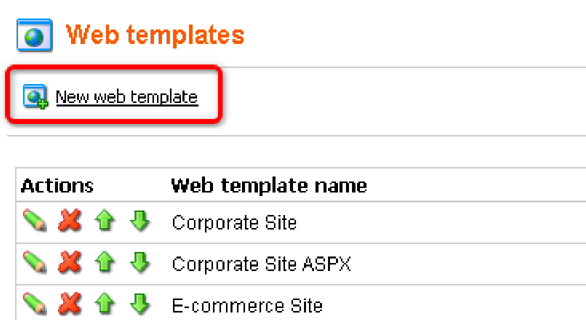
1) Export your site. For a step-by-step tutorial on how to export a site, please refer to the Exporting a site chapter of this guide.

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.



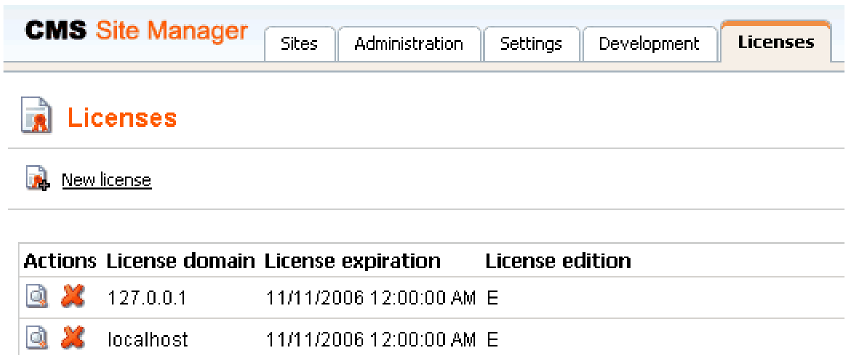

5) Enter the following details:

- **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 field.

### How licensing works

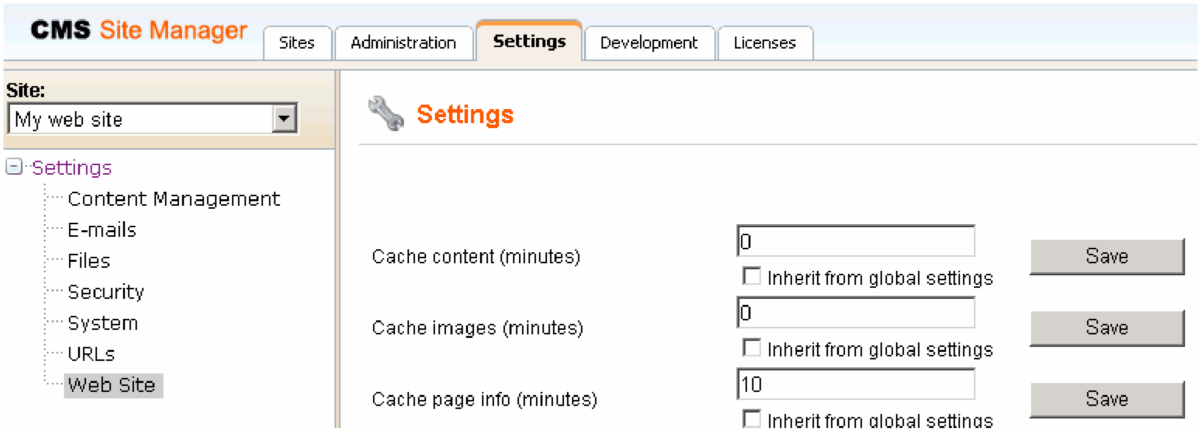
If you're running web site on domain mydomain.com, you need a single license key that will also work for:

- http://mydomain.com
- https://mydomain.com
- http://www.mydomain.com
- https://www.mydomain.com
- http://localhost
- http://127.0.0.1

If you use a domain alias (domain name that points to the same web site), such as mydomainX.com or mydomain.net, you also need license keys for these domain aliases. You need to 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

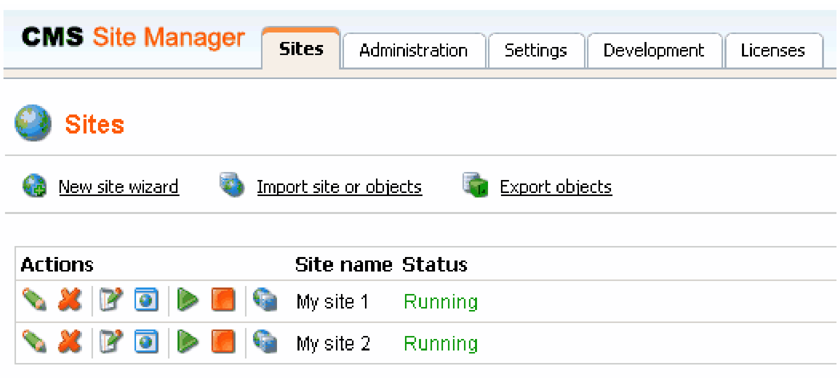
Kentico CMS allows you to run multiple web sites from the single installation (code base) and database. All web sites run as a single web site in IIS. The following tutorial explains how to set up multiple web sites on Windows XP and Windows Server 2000/2003.

We will configure two web sites:
- **mysite.com**
- **mysite2.com**

### Configuring multiple sites in Kentico CMS (common for all operating systems)

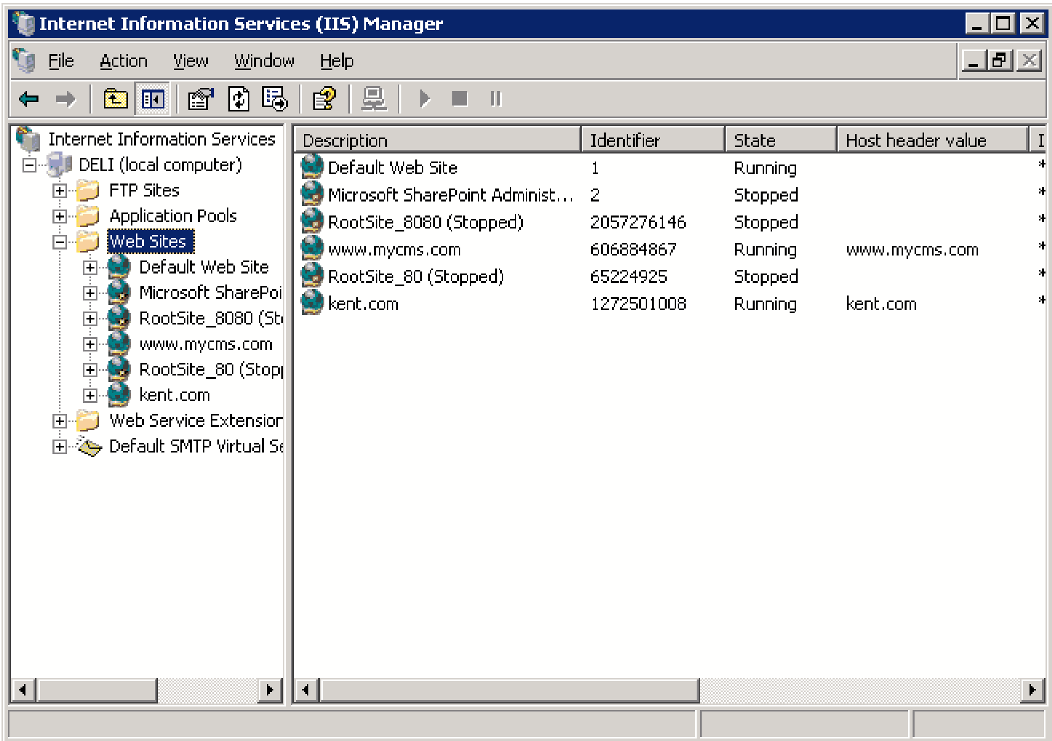
This part is common for all operating systems.

1. Create two web sites in **Site Manager -> Sites** - you can either import your existing web sites or you can create new web sites using the New site wizard.
2. Edit the properties of each web site in **Site Manager -> Sites** and set the **Site domain name** value of the each web site 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**.

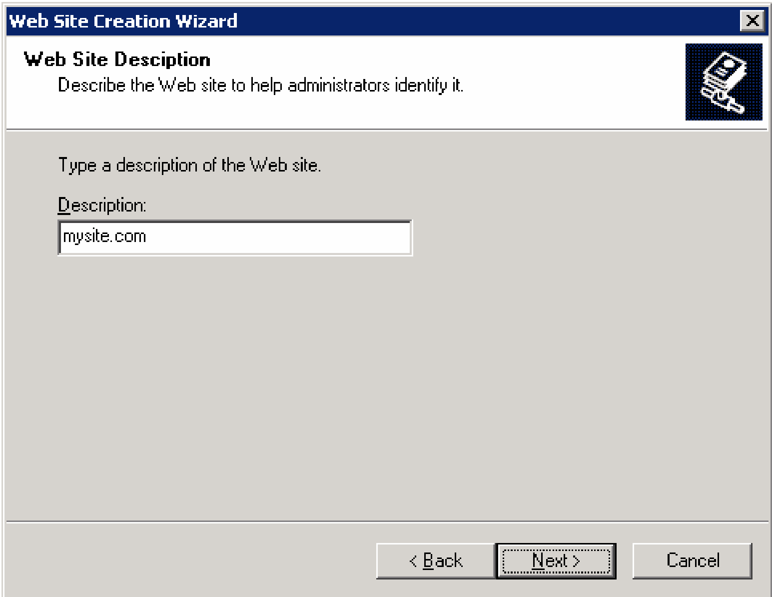


## Configuring multiple sites on Windows Server 2000/2003

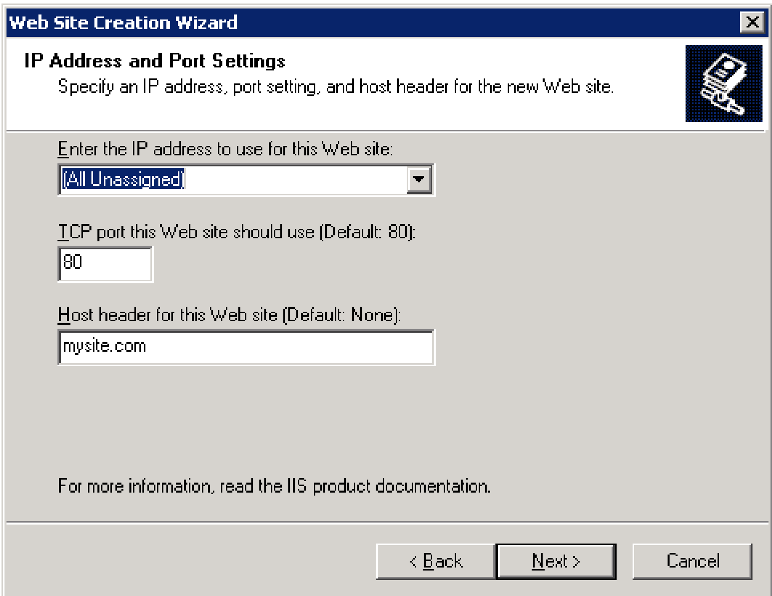
Open **Start -> Administrative tools -> Microsoft Internet Information Services (IIS) Manager**, go to Web sites section and if the web site does not exist yet, create it.



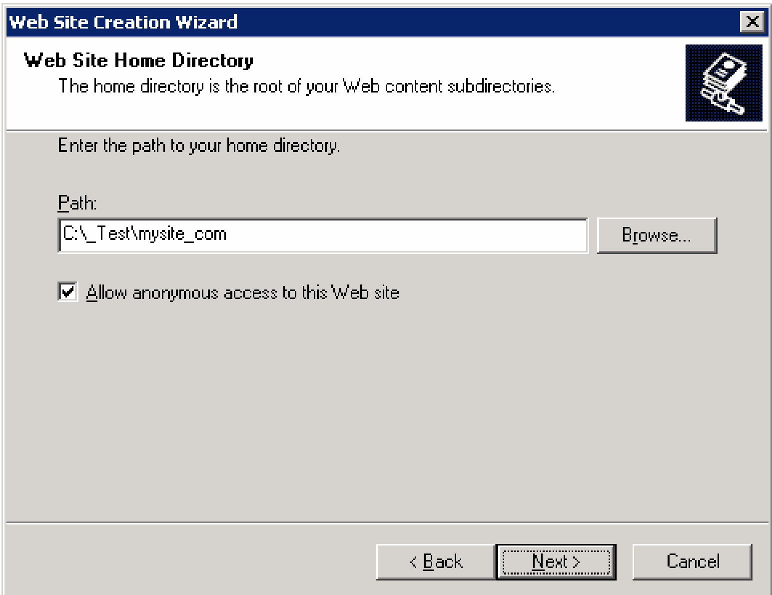
Right-click **Web sites** and choose **New -> Web site...** The Web Site Creation Wizard starts. Enter the site descriptive name, such as mysite.com:



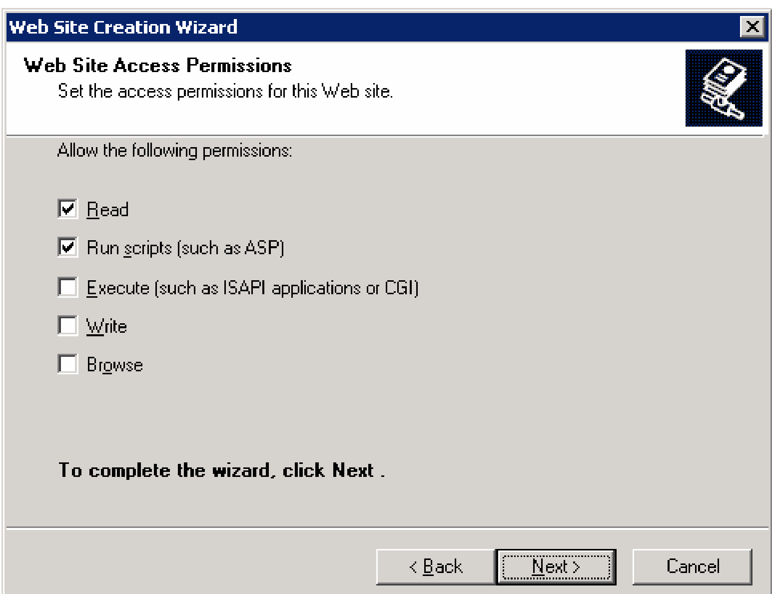

Click **Next**. Enter the **Host header for this web site** as mysite.com:



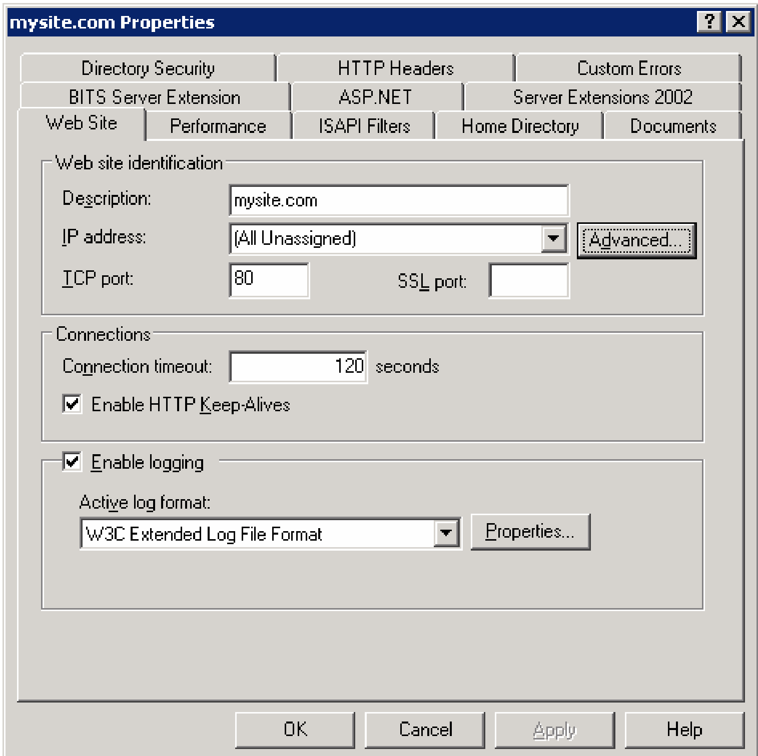
Click **Next**. Select the disk path where the root of your web site is placed. This must be the folder where web.config file is placed.



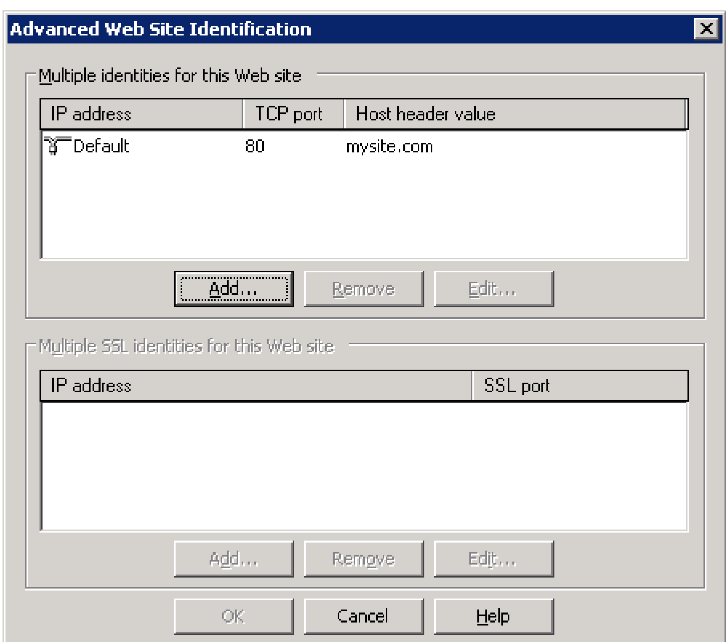

Click **Next**. In the next step, make check the **Read** and **Run scripts** boxes.



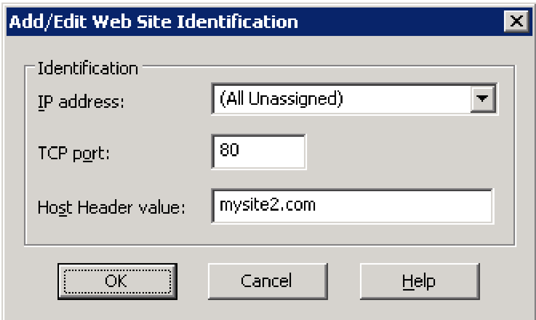
Click **Next**. The wizard is finished. Click **Finish**. Now we need to add the other domain name to the list of URLs hosted by this web site. Right-click the newly created web site and display web site properties. On the **Web Site** tab, click **Advanced...**



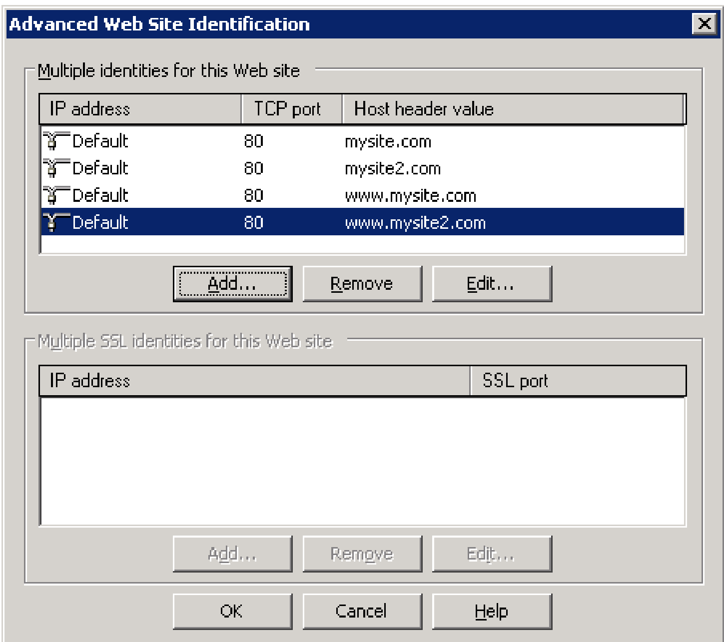

Now you need to add another host header value for your second domain:



Click the top **Add...** button and enter the appropriate values. The standard HTTP port is 80:

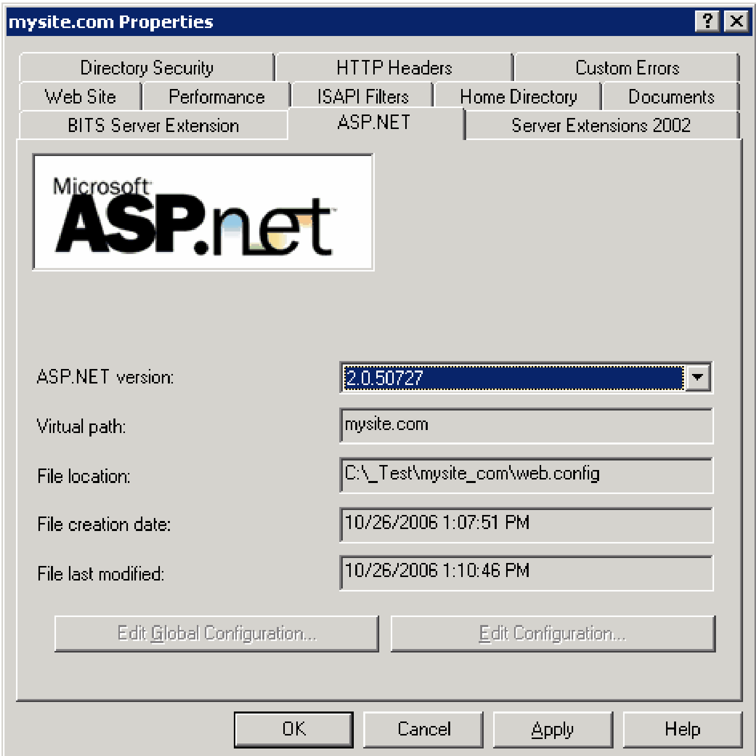



You need to repeat this also for www.* alternatives of your web site:




Click **OK** on all dialogs.

**Please note:** You may also need to configure the web site for ASP.NET 2.0 in web site Properties, on the **ASP.NET** tab:




Your new web site 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 web site.

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:

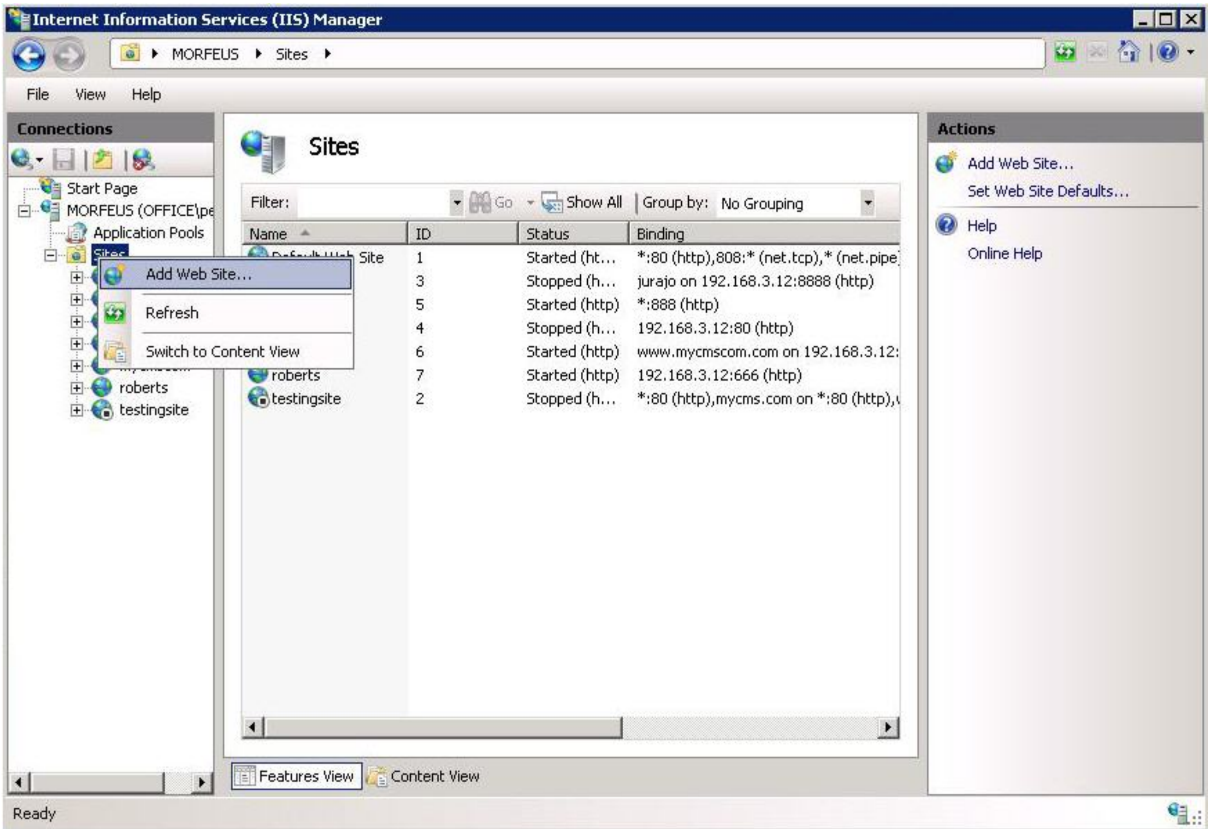
```
127.0.0.1      mysite.com
127.0.0.1      www.mysite.com
127.0.0.1      mysite2.com
127.0.0.1      www.mysite2.com
```

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 web sites.

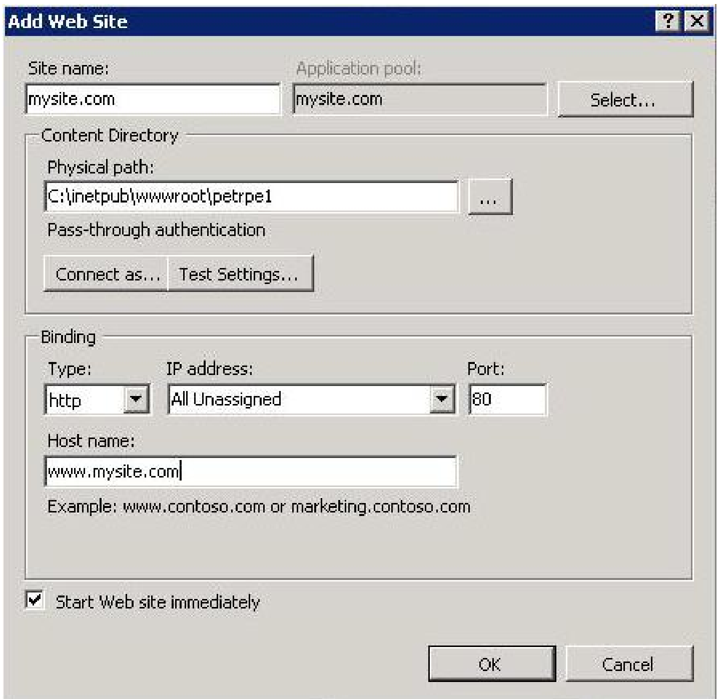
## Configuring multiple sites on Windows Vista / Server 2008

Go to **Start -> Control Panel -> Administrative tools -> Microsoft Internet Information Services (IIS) Manager.** In the tree view, right click **Sites** and choose **Add Web Site...**.

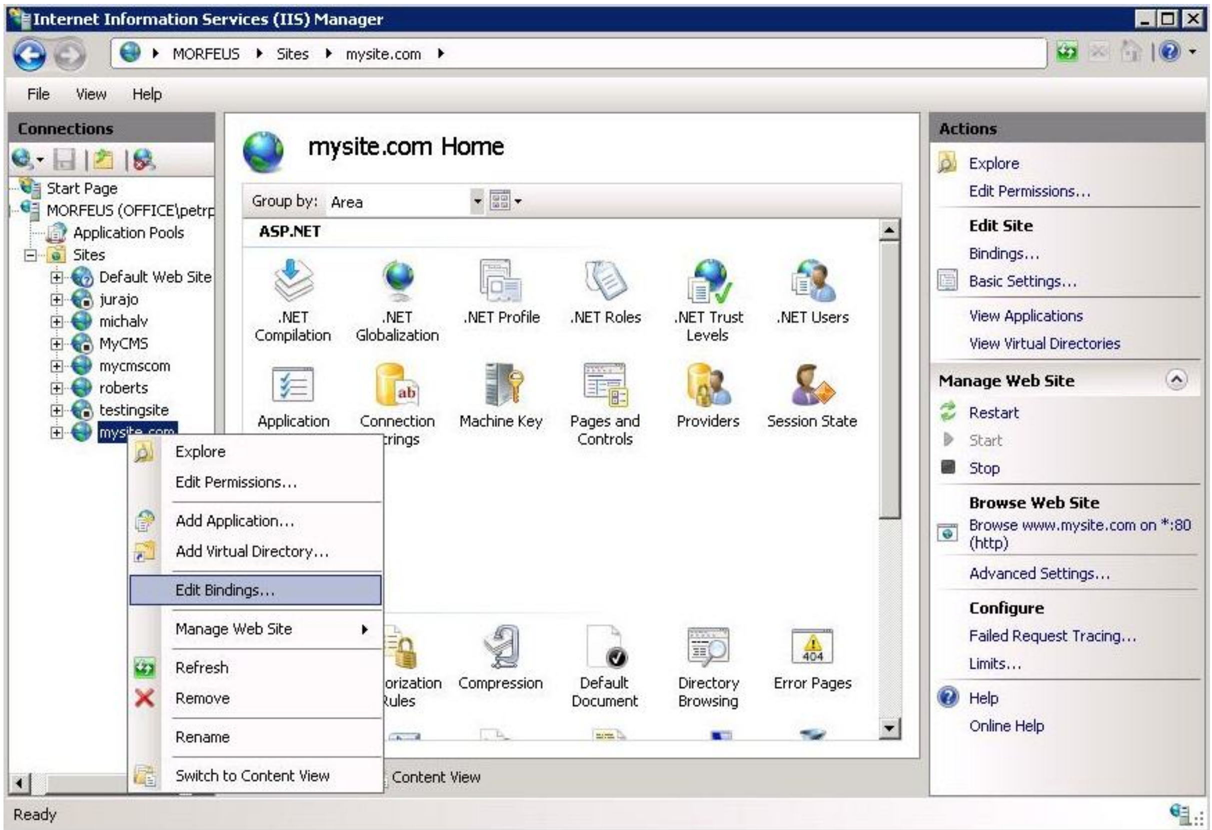


Enter the following details and click **OK**.

- **Site name**: mysite.com
- **Physical path**: disk path to the location where your web site is placed; this must be the location where the *web.config* file is stored
- **Host name**: www.mysite.com



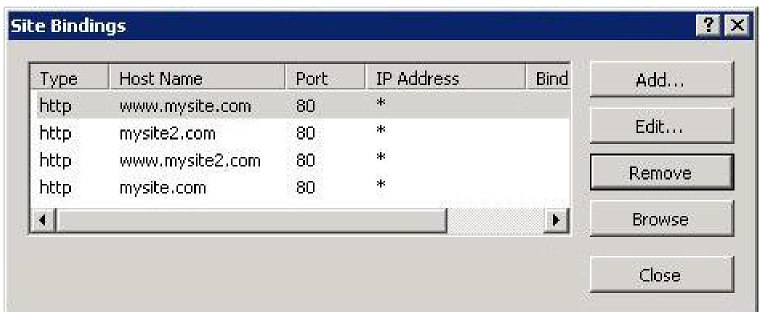
The site should now appear in the tree, under the **Sites** node. Right click the site and choose **Edit bindings** (or **Bindings** on Vista).




The site bindings dialog appears. Click **Add...**.

Enter the domain name of your second web site (*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 web site 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 web site.

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 web sites.

### 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 web site 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 web site) and to http://www.mysite2.com. You should see two different web sites.

**Multiple web sites 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 web sites by subfolder (virtual directory). Read chapter Multiple web sites on a single domain (in subfolders) for more details.

**Using multiple web sites on Windows XP**

Windows XP allows you to run only one IIS web site at a time. If you need to develop multiple web sites (multiple Kentico CMS instances) in the root folder, you may need to create additional web sites 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 web sites in separate subfolders, without getting a new domain.

In this case, you can configure the domain names (or domain aliases) of the web sites like in the following example:

### Example

1) Install Kentico CMS on the main domain and make sure the web site works correctly. Create another web site in the same installation. The following steps assume that your installation URL is **http://localhost/mykenticoweb** and the web site files are stored in folder **C:\inetpub\wwwroot \mykenticoweb**.

2) Configure web site 1 domain name as: **localhost/mykenticoweb/web1**

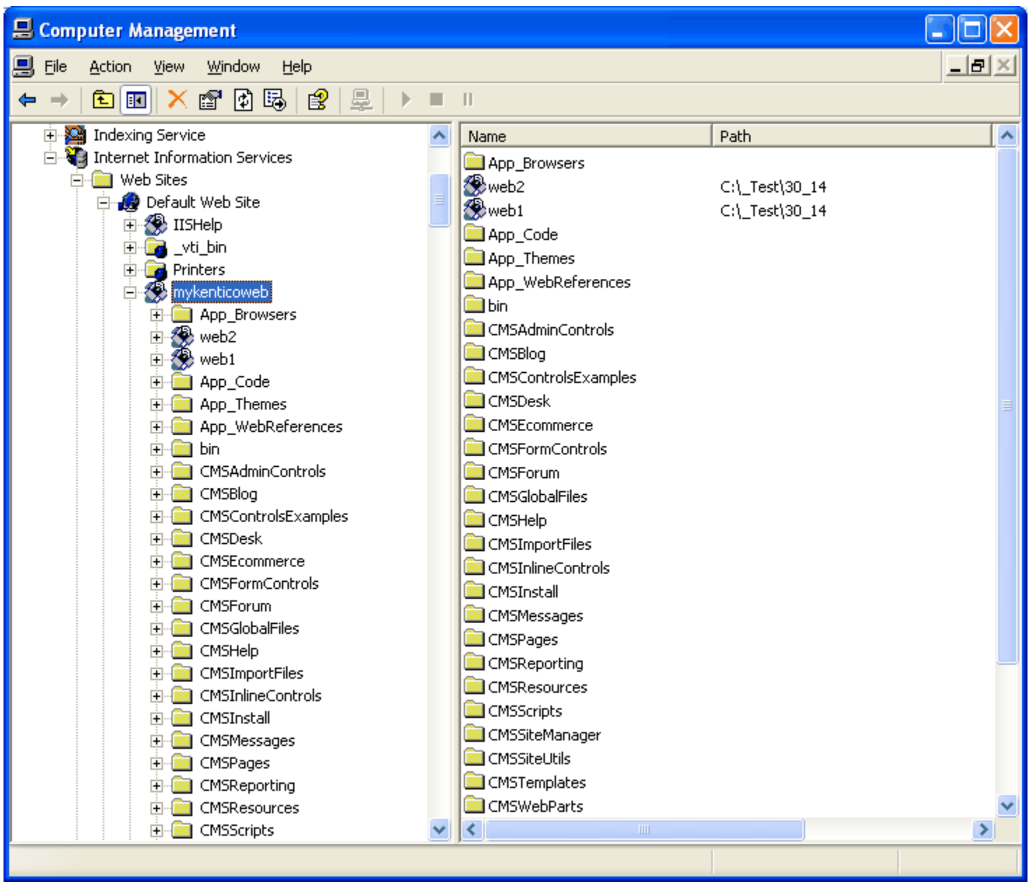
3) Configure web site 2 domain name as: **localhost/mykenticoweb/web2**

4) Open IIS console (Control Panels -> Administrative Tools -> Internet Information Services) and locate the web site and the virtual directory **mykenticoweb**.

5) Create two new virtual sub-directories under mykenticoweb called **web1** and **web2**. Both of them will use the root of your mykenticoweb web site on the disk, which is **C:\inetpub\wwwroot \mykenticoweb** in this example.

Now when you go to **http://localhost/mykenticoweb/web1**, you will see web site 1. If you go to **http:// localhost/mykenticoweb/web2**, you will see web site 2.

The following figure shows the configuration in the IIS console:



## 5.12 API Examples

You can use Kentico CMS API to create new web sites or update and delete existing ones. To create a new web site, you need to import some of the web site templates (typically the Blank web site template stored in folder CMSSiteUtils\Templates\BlankSite.

You can find API examples in chapters Import and export of the web site, Update web site properties and Delete web site.

# Part VI

# 6 Content management

## 6.1 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.

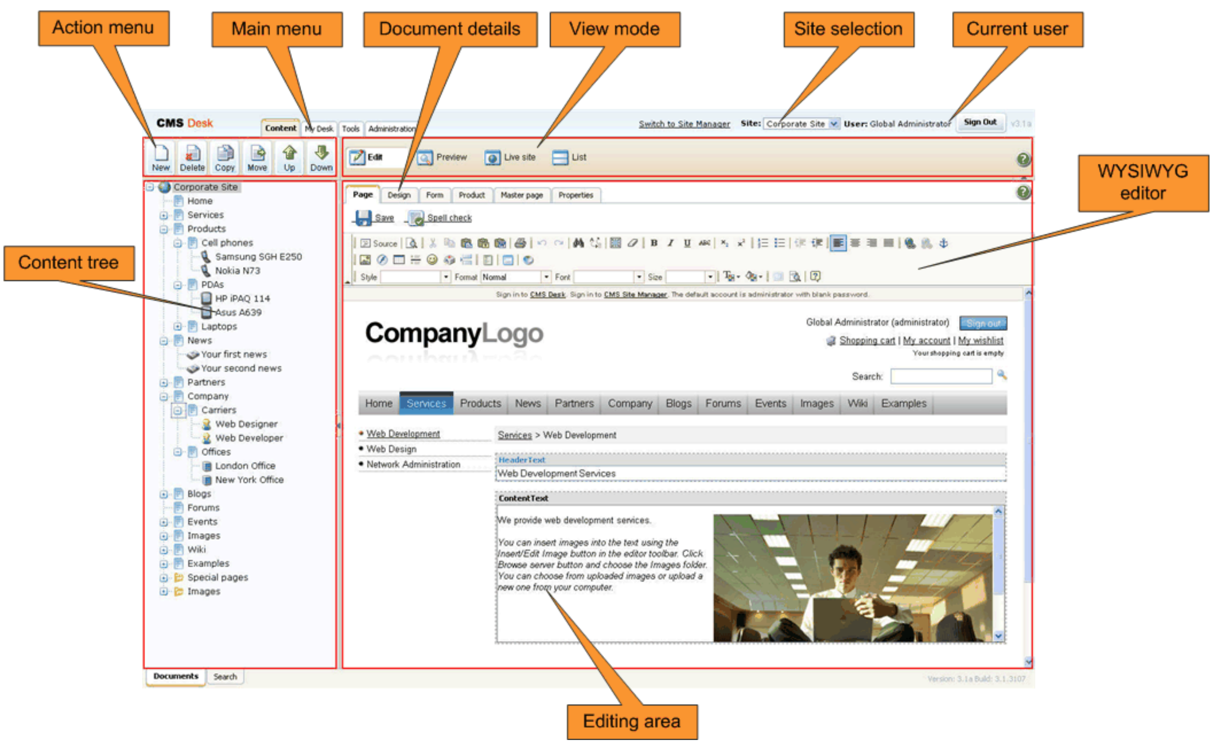
**Default user name and password**

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 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.
**Please note:** the addresses may be different depending on your installation and domain name.

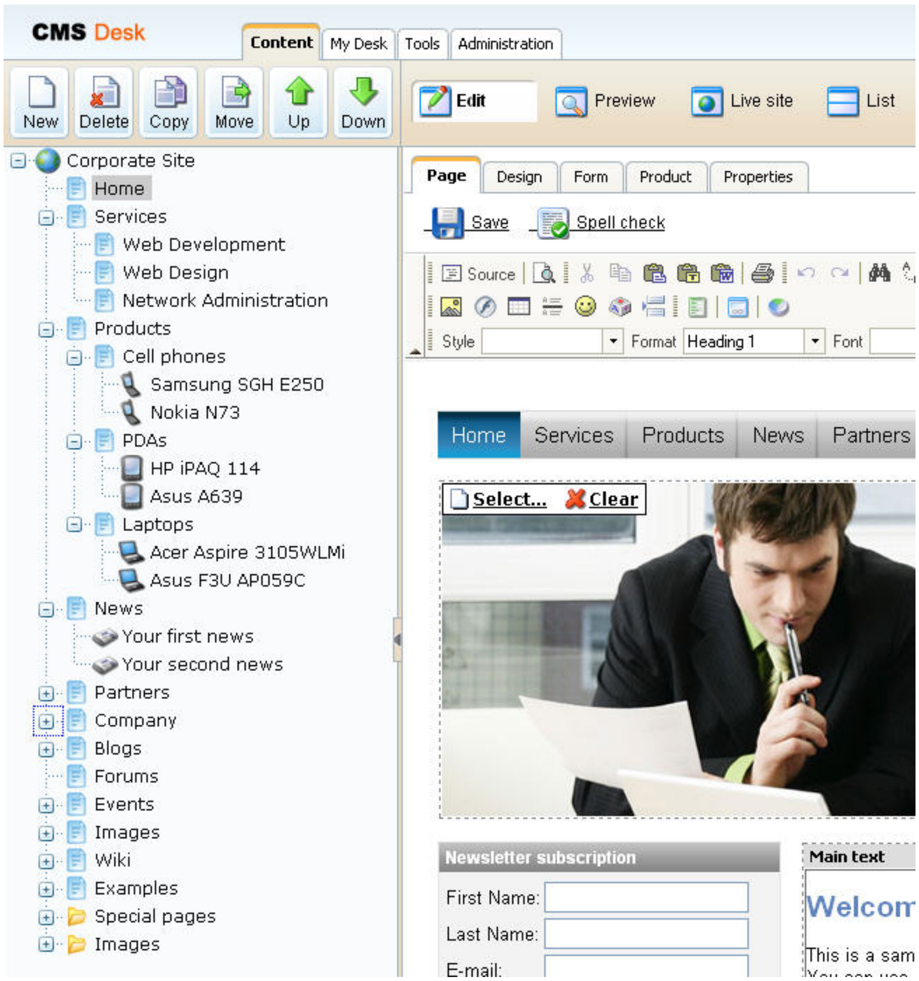
The following figure shows the CMS Desk user interface.




See also: Where is the content stored?, How do I edit content?

## 6.2 Organizing pages, files and documents

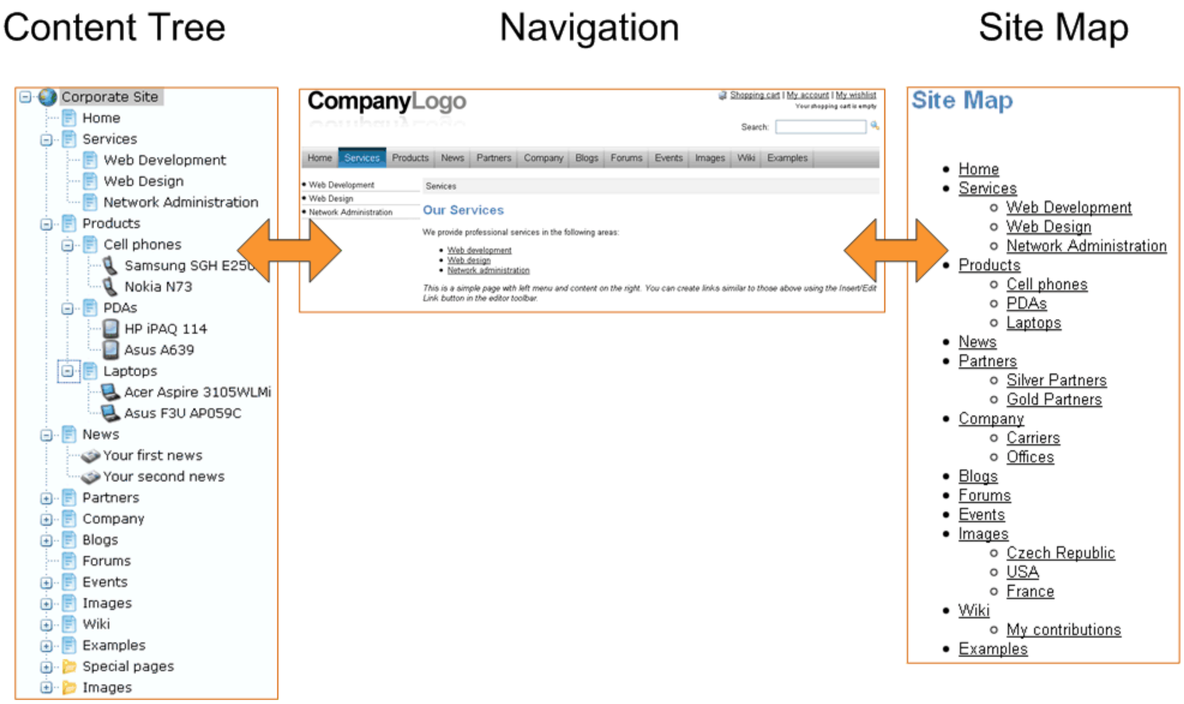
All content in Kentico CMS is stored in a **tree hierarchy**. You can see the content tree in **CMS Desk -> Content**:




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 web site:




## Document types

Each document is of some type - it can be a page with unstructured text, news item, article, press release, product specification, etc. The document types can be customized with your own fields.

## 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.<br><br>Generally: pages with simply structured or unstructured, text-only content.<br><br>The editable regions are usually used only in connection with documents of type **Page (menu item)**. | News, product specification, event details, job opening, etc.<br><br>Generally: pages with structured content where you need to separate content from design and keep the content in its original data type.<br><br>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 web site:

- **Web site design files -** Images and Flash files that are used on the web site 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 web site 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.

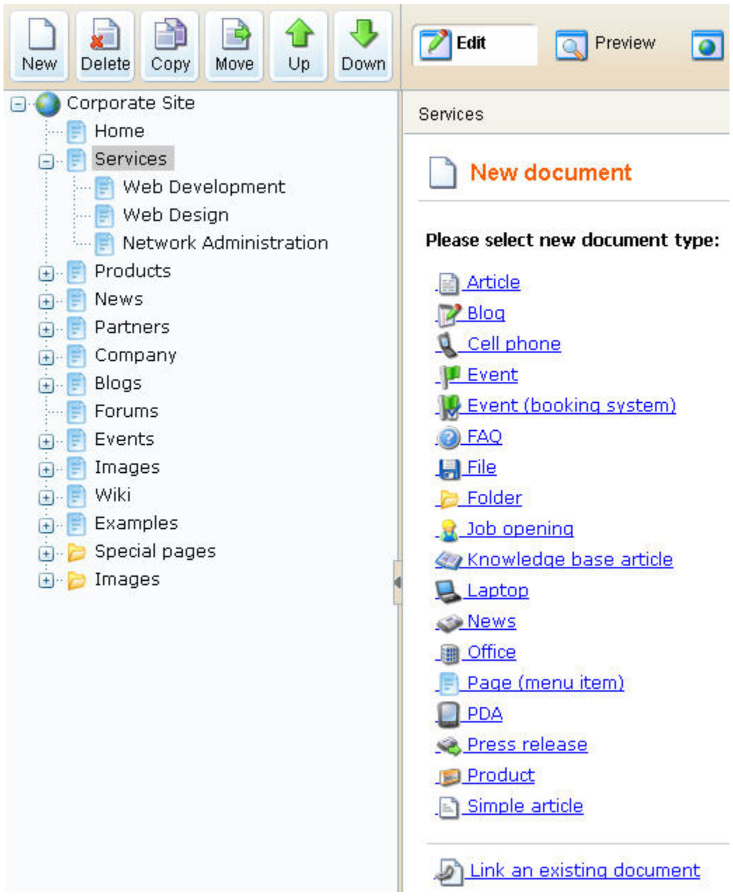
You can find more details on files in chapter File management.

See also: Where is the content stored?

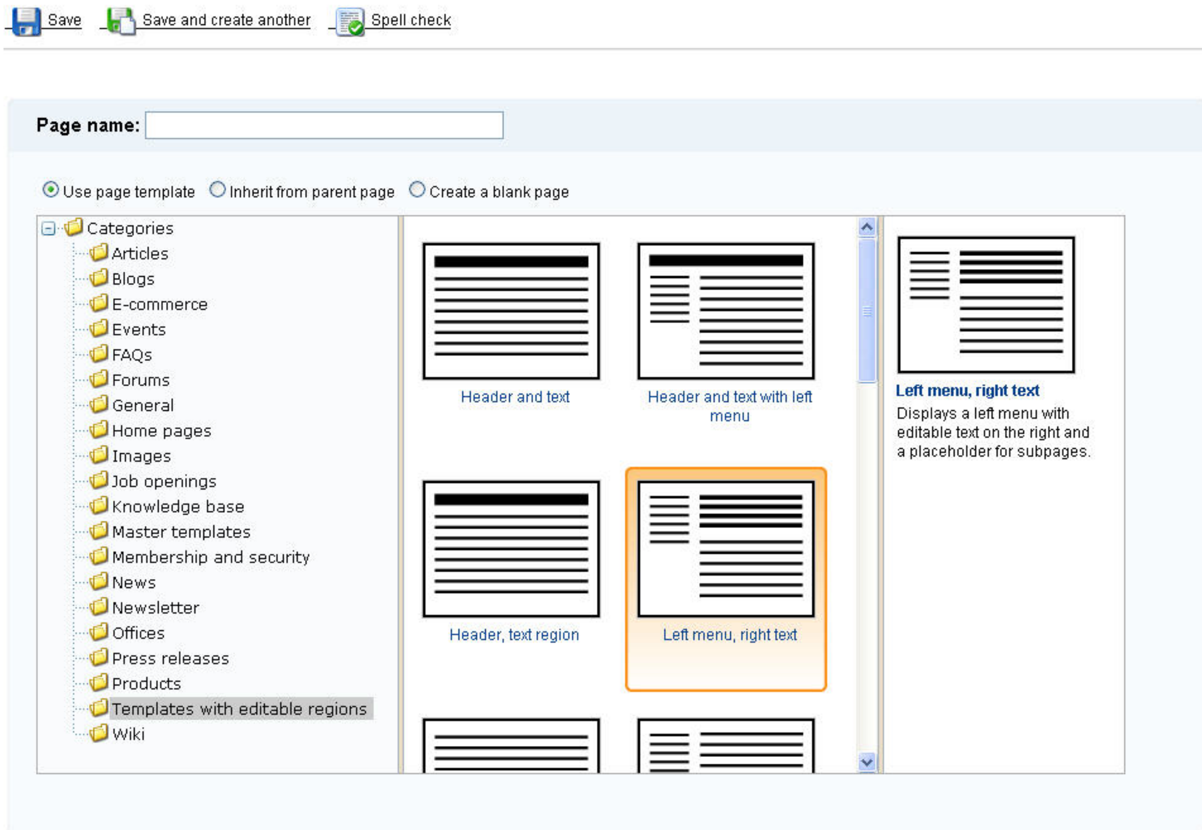
# 6.3 Editing content

## 6.3.1 Creating a new page

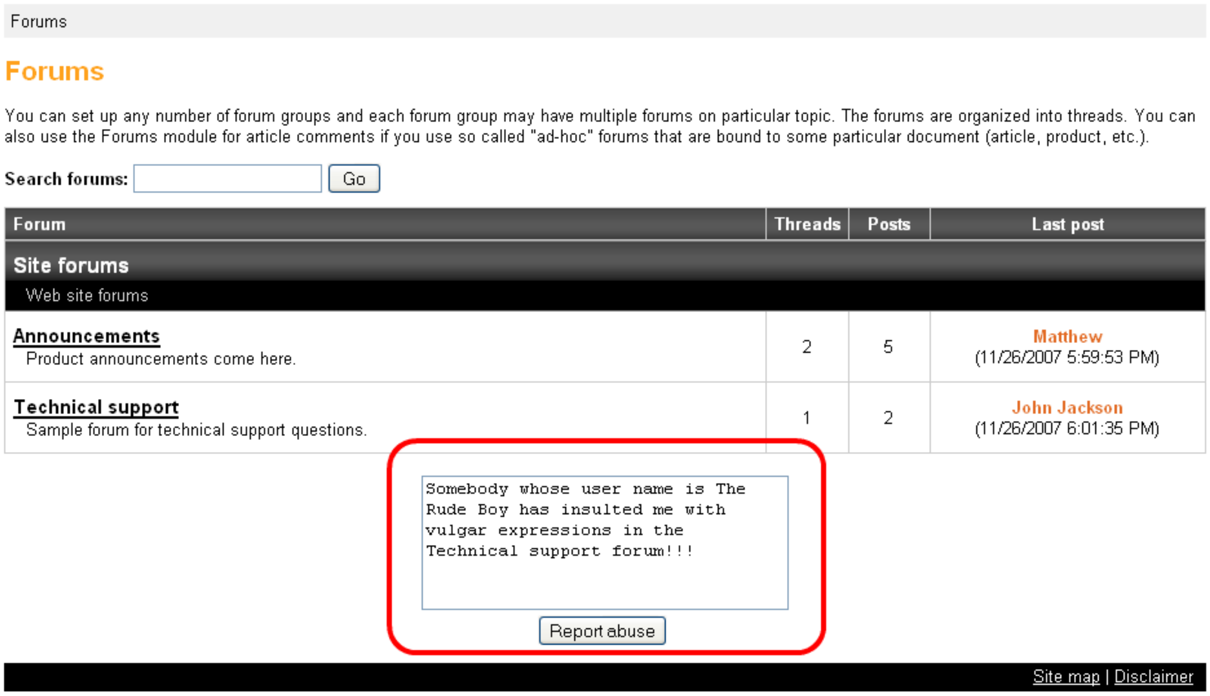
When you're creating a new page, you need to click the **New** button in the main toolbar and choose to create a new document of type Page (menu item):



Then, enter the **page name** and select the **page template**. The page template defines the layout and design of the new page:



Click **Save** to create a new page. You're redirected to the Page tab and you can enter some content. Click **Save** to save the changes.



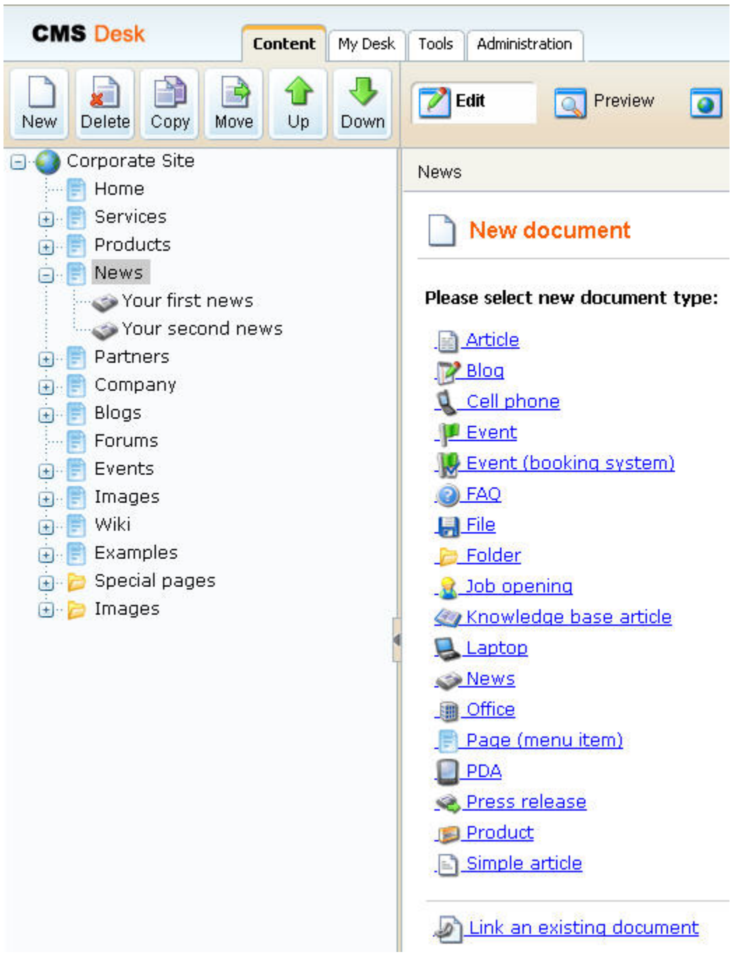

You can alternatively choose one of the following options when creating a new page:

- **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.

### 6.3.2 Creating a new structured document

The following chapters will explain how you can edit the content.

You can create new documents in **CMS Desk -> Content.** Click the document under which the new item should be placed, click **New** and choose the type of the document:



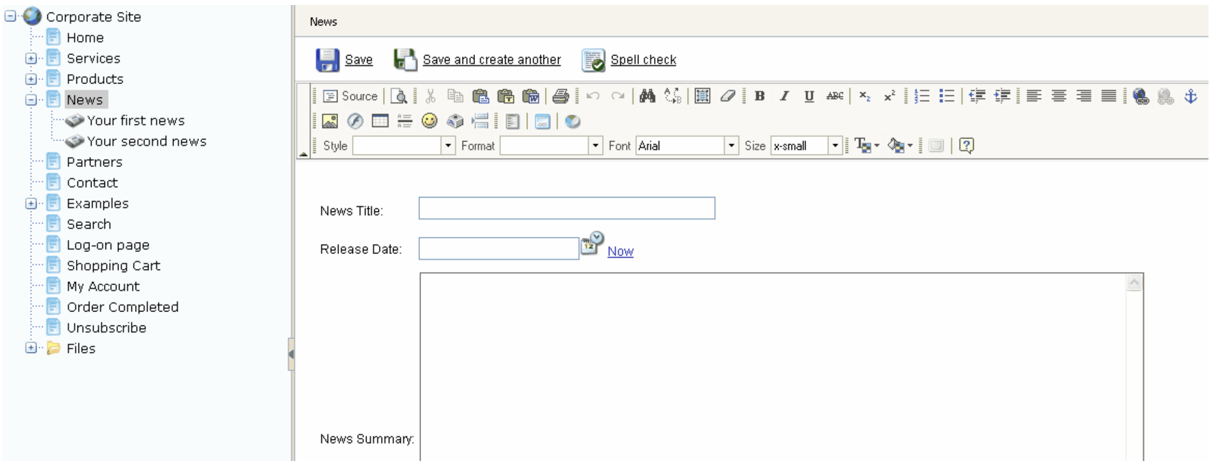

You are then redirected to the appropriate editing form.

**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.

**Example:**

Click **News** in the content tree, click the **New** button and choose to create a new **News** document. You will be displayed with editing form like this:



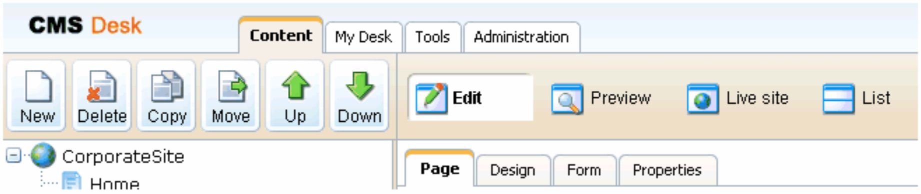

Enter values like these:

- **News title:** My testing news
- **Released date:** 8/15/2007 (the release date displayed on the web site)
- **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 or you can 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 enter several documents of the same type in the same location at once.

### 6.3.3 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.

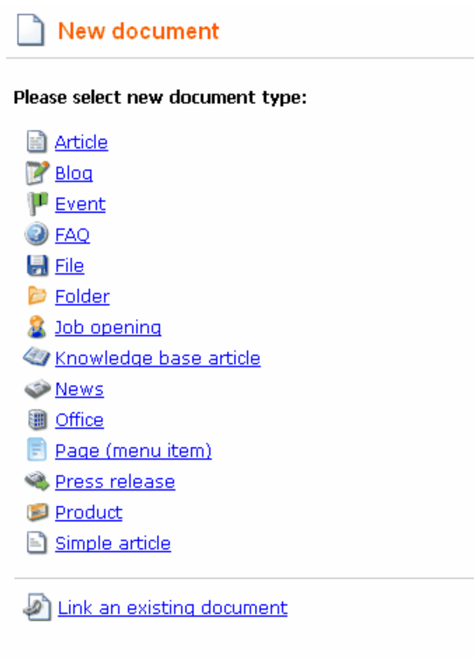
**If you do not use workflow** for the selected document, the **Preview mode** displays basically the same content as the **Live site** mode, the only difference may be caused by the fact that the preview mode doesn't use caching, so it may display also changes that are not visible on live site yet due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

## 6.3.4 Creating a linked document

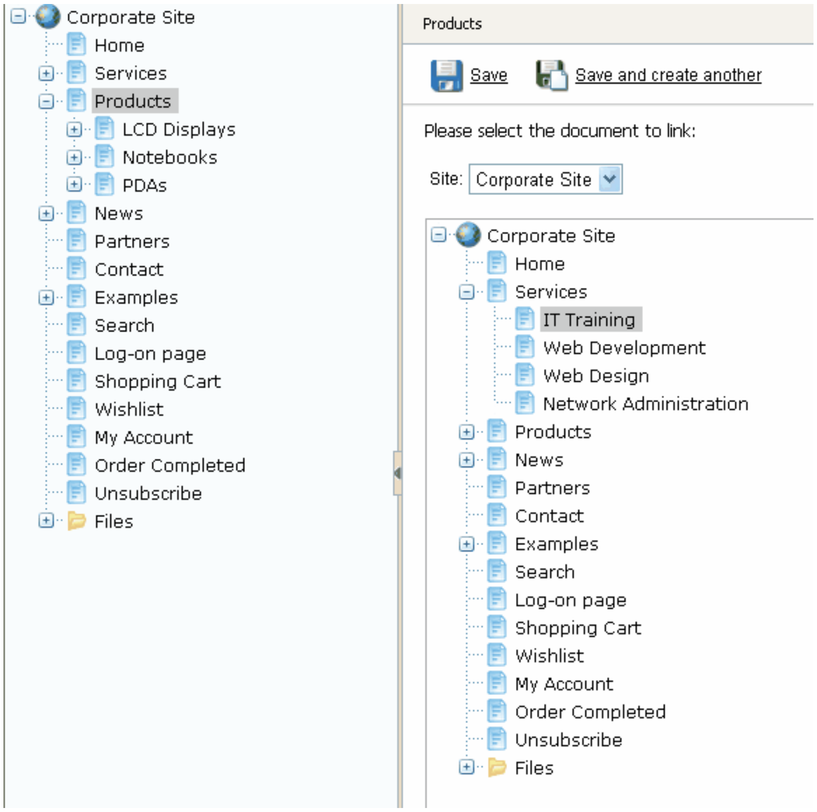
A linked document is a "stub" or "shortcut" of some existing document. It allows you to place a single document to multiple places in the content tree instead of creating its copies. Such a document is then displayed in the given part of the web site, but when you edit it, you actually update the original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

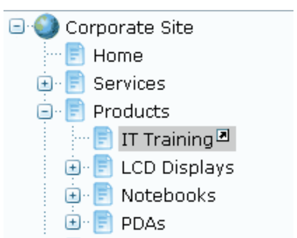
In order to create a linked document, click **New** and choose **Link an existing document** in the bottom of the screen:



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:

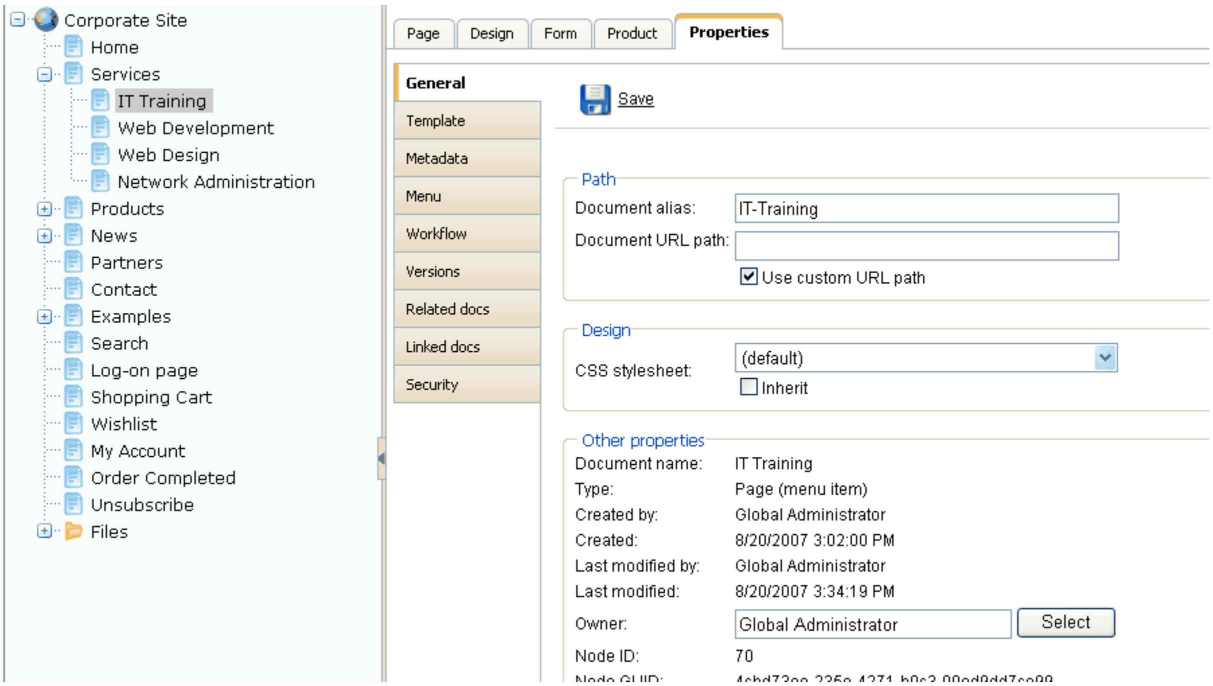



You can see the list of all linked documents for the currently selected document in the Properties -> Linked docs dialog.

# 6.4 Document properties

## 6.4.1 Overview

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




### Customizing the Properties Dialog

The Properties tab and its sub-sections can be optionally hidden from chosen roles. It allows you to customize and simplify the look of the editing interface.

You need to enable personalization in **Site Manager -> Settings -> Content management -> Personalize user interface by permissions**. Then, you can customize the user interface by configuring permissions for module **CMS User Interface** in **CMS Desk -> Administration -> Permissions**.

### 6.4.2 General

General tab fields:

| | |
|---|---|
| CSS Stylesheet | The CSS stylesheet used for the given 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.<br><br>This feature doesn't imply any special permissions for the owner, but it allows for easier orientation in documents. The owner can see all her documents in **My Desk -> My Documents** section.<br><br>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. |
| 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. |

### 6.4.3 URLs

URLs tab fields:

| | |
|---|---|
| Document alias | The unique name of the document in the given section of the web site. This name doesn't change when you modify the document name and it's used:<br>1. to define the AliasPath, which is the unique path of the document. The developers typically use the AliasPath in the Path property of the web parts and controls.<br>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.<br><br>**Forbidden characters**<br><br>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:<br>**/cms-for-asp-net**<br>The URL path must always start with /. Then, the document is accessible through two URLs:<br>www.kentico.com**/products/cms**.aspx<br>www.kentico.com**/cms-for-asp-net**.aspx<br><br>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.<br><br>The URL path is, unlike the alias path, culture-specific. It means if you use URL paths:<br>/product (for English version)<br>/produkt (for German version)<br><br>then when the user comes to /produkt.aspx, the language of the web site is automatically switched to German. |
| URL extensions | Extensions supported for this document. Multiple extensions can be entered divided by a semicolon. If you enter a semicolon without any extension, extensionless URLs will be allowed.<br><br>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.<br><br>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. 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. |

## 6.4.4 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 on the **Select** button. |
| Save as new template... | After editing default template, modified template can be saved as a new one, so that original template remains unchanged. |
| Inherit template | Current page inherits template from its parent document. |
| Clone as ad-hoc template | Creates ad-hoc template which you can modify without the need of creating new template and fear of modifying the original one. The ad-hoc template can be used only for a given page. If you need to use the same template on some other page, create a new template. |
| Edit template properties | Display a new window with the template properties and settings. |

### Inherit content

This option allows you (developer) specify if 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 the parent document. |
| Do not inherit any content | Document does not inherit any content from the parent page. |
| Select inherited levels | Select document at given level from which should be the content inherited. |

## 6.4.5 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 web site. |
| Page keywords | Keywords used for meta tags of the page and for searching content of the web site. |

**Global Settings**

You can configure prefix of the page title, description and keywords for all documents on the web site using the Site Manager -> Settings -> Web Site 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.

## 6.4.6 Categories

On this tab, you can assign the document to categories. These are topic-related groups of documents. See chapter Module Categories for more details.

## 6.4.7 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).<br><br>**Please note:** the document is displayed in the navigation if all of the following conditions are met:<br><br>1. The Show in navigation box is checked.<br>2. The document is published.<br>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.<br>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).<br><br>**Please note:** the document is displayed in the navigation if all of the following conditions are met:<br><br>1. The Show in site map box is checked.<br>2. The document is published.<br>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.<br>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.<br>- example: alert('You are being redirected outside our web site.'); |
| URL redirection | The user is redirected to the target location when the menu item of this document is clicked.<br>- example: ~/products/overview.aspx - redirects user to the given page |

**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 | CSS style, such as *color:red* |
| Menu item CSS class | CSS class name defined in the CSS stylesheet, such as MyMenuClass |
| Menu item left image | Image displayed on the left side of the menu, such as ~/App_Themes/Green/Images/bullet.gif |
| Menu item image | Image displayed instead of the menu text, such as ~/App_Themes/Green/Images/bullet.gif |
| Menu item right image | Image displayed on the right side of the menu, such as ~/App_Themes/Green/Images/bullet.gif |

## 6.4.8 Workflow

Here 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 chapter Workflow and versioning for more details.

## 6.4.9 Versions

Here you can view the previous versions of the document if it uses workflow. You can also roll back to a previous version of the document, delete the older version or destroy the whole document history.

See chapter Workflow and versioning for more details.

## 6.4.10 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

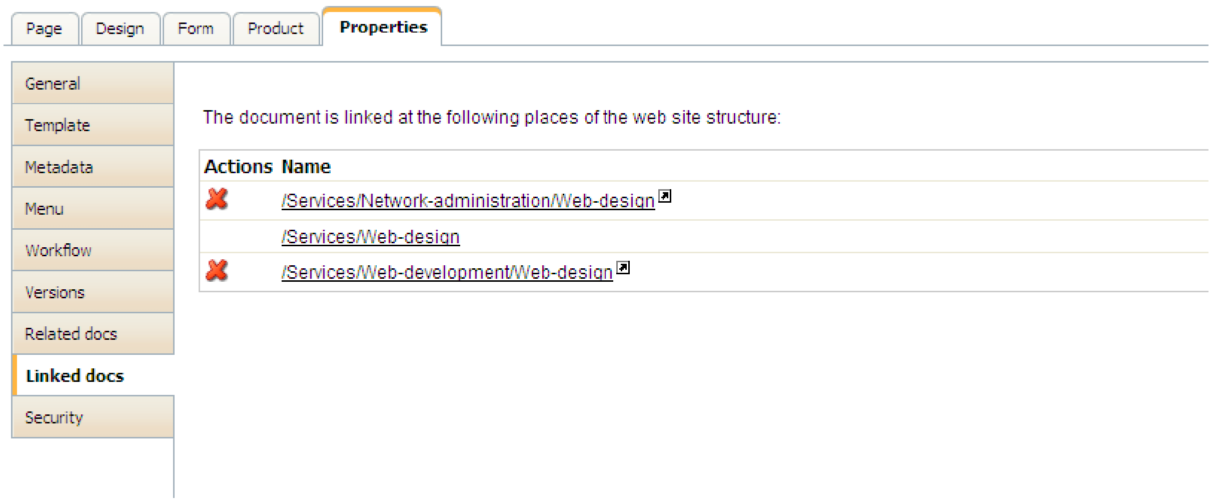

The related documents need to be displayed by a control (web part) that is configured for displaying related documents. If you're a developer, please see the **Kentico CMS Web parts and Controls reference -> Displaying related documents** for more details.

The relationship names need to be defined in **Site Manager -> Development -> Relationship names**. Only relationship names enabled for the current web site are available in the drop-down list.

## 6.4.11 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 cross icon or navigate to other linked documents or to the original document by clicking the link:




## 6.4.12 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 to roles in dialog **CMS Desk -> Administration -> Permissions** 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 web site areas chapter for more details.

You can also specify if the give page or web site section **requires SSL** (HTTPS) protocol. If you set it to Yes, the CMS automatically redirects the user to the URL with http<u>s</u>:// protocol.

# 6.5 Document types

Each document is of some type. Document types are fully customizable - you can add, modify and delete custom fields. Please read more details in chapter Document types.

# 6.6 File Management

## 6.6.1 File management overview

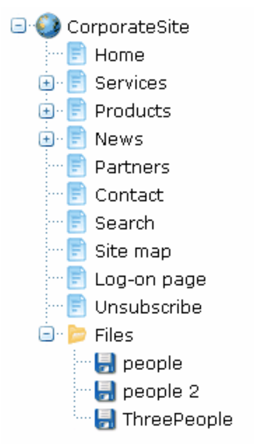
Kentico CMS allows you to upload media files (such as GIF, JPG, SWF, PDF, XLS, DOC, etc.) to the Kentico CMS database or file system and manage them as any other content.

### File management approaches

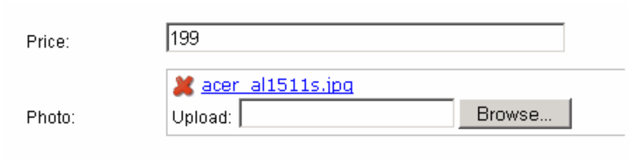
There are three types of media files from the file management perspective:

- **Unmanaged files** - these files are part of the web site 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.

- **Managed, stand-alone files** - these files are uploaded by the content editors as new documents in 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 the editable regions of the page. You may use documents of type *Folder* to organize the uploaded files:



- **Managed, document-bound files -** 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 will typically use this type of files for structured content, such as product image uploaded to the product details form. When the document is deleted, the attached files are also deleted.




## 6.6.2 Where the files are stored

The files can be physically stored in the file system, in the database or in both:

- **File system** - the files are stored in the configured folder on your disk. This option provides the best performance, however, your web application must be granted with Modify permissions on the disk which is not always possible.

- **Database** - the files are stored in the database. This option provides a worse performance, but it allows you to use full-text search in uploaded files, it doesn't require Modify permissions on the disk and it allows you to easily backup the uploaded files as a part of your database backup.

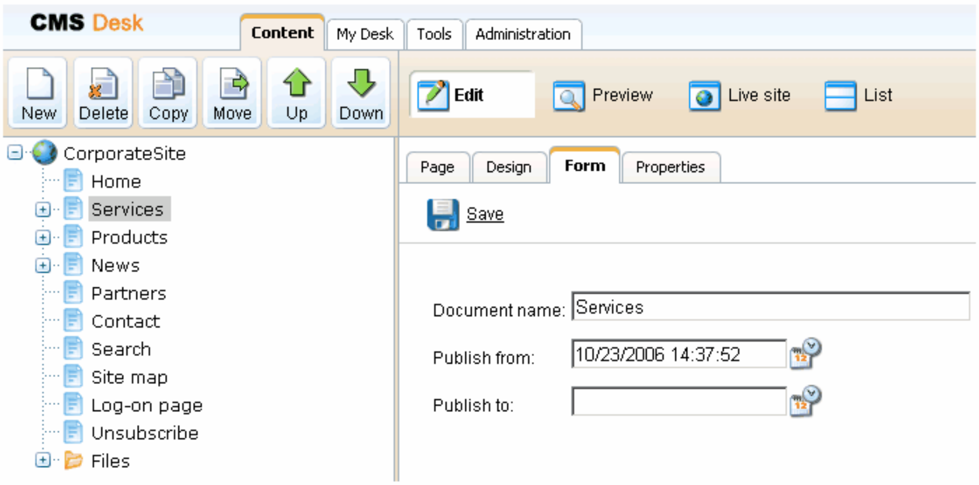
- **Database and file system** - this option combines the advantages of both options. It provides the same performance as the file system-only option since the files are stored on the file system as well. At the same time, you can use workflow, versioning and full-text search.

You can configure the file storage in **Site Manager -> Settings -> Files**. Here you can configure the following options:

| | |
|---|---|
| Files folder | The folder on the disk where the files are stored. You can use either the physical disk path or virtual path like ~/UploadedFiles.<br><br>If you do not specify any value, the files are stored in folder ~/<site code name>/files. |
| Generate thumbnails | Indicates if the CMS should generate 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 the performance**. |
| Redirect files to disk | Indicates if the user should be redirected to the file on the disk. This option provides the best performance. It requires that the **Store files in file system** box is checked. |
| Store files in database | Indicates if files should be stored in the database. |
| Store files in file system | Indicates if files should be stored in the file system. |
| Upload extensions | Allowed file extensions for the uploaded files. You can restrict the types of uploaded documents to e.g.: gif;jpg;doc;pdf<br><br>It allows you to block users from uploading potentially dangerous files, such as ASPX scripts. |

# 6.7 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 web site 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 web site and search for the **Content scheduling** task.

# 6.8 WYSIWYG editor

## 6.8.1 Overview

Kentico CMS comes with built-in WYSIWYG editor. It uses the FCKeditor, one of the best browser-based editors available on the market. Since the editor is fully integrated in the CMS, it cannot be replaced with any other editor.

## WYSIWYG editor toolbar

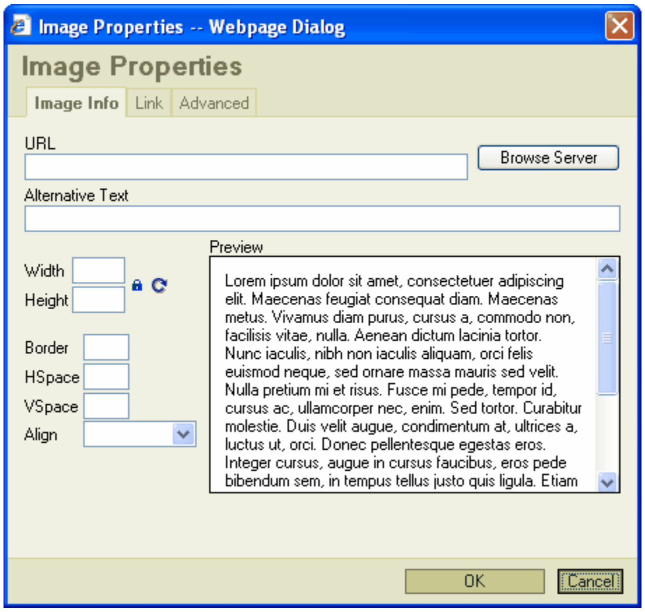
The default toolbar looks like this:


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

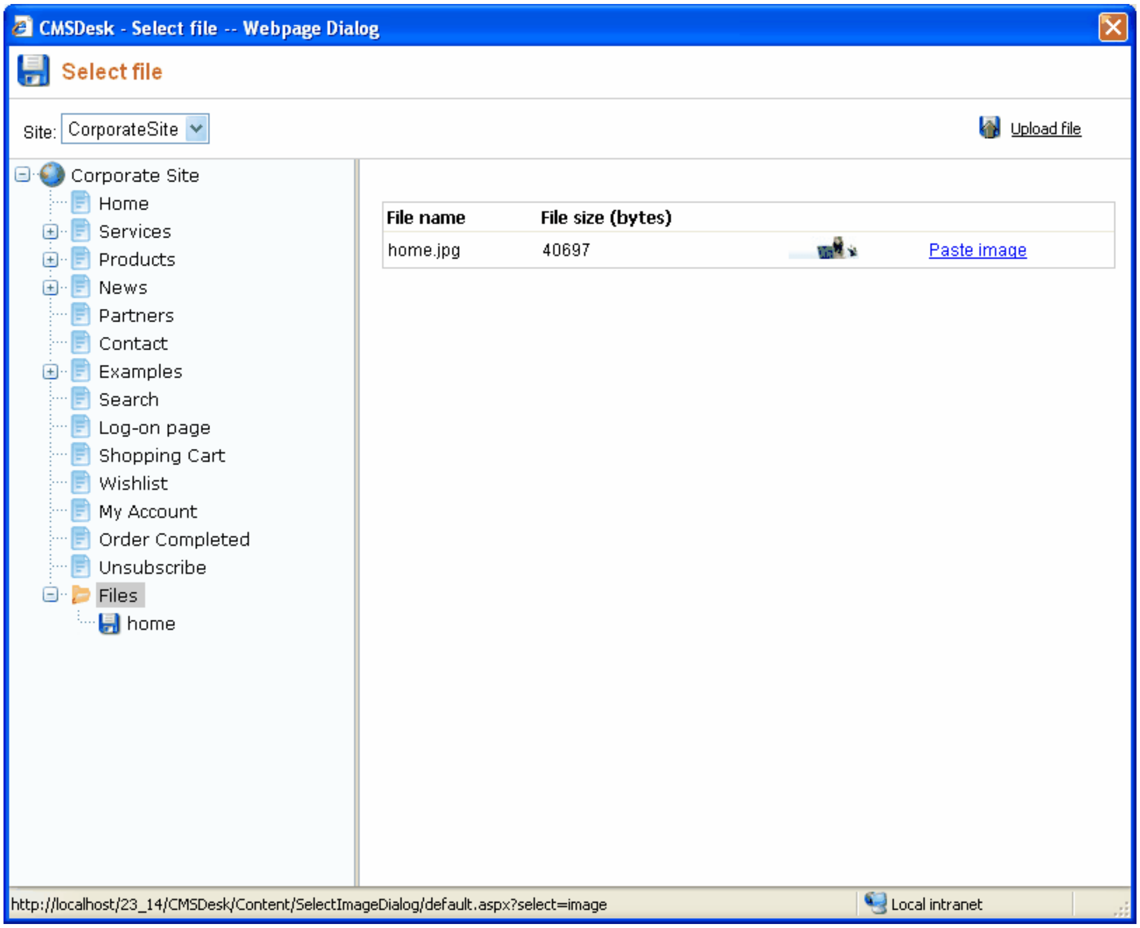
| | |
|---|---|
|  | Paste as plain text - this command pastes the content of your clipboard while cleaning out all formatting. |
|  | Paste from Word - this command allows you to clean the content pasted from Microsoft Word. It cleans up the HTML code so that it doesn't contain extra code and so that it fits your web site design. You can find more details in chapter Copy & Paste from Microsoft Word. |
|  | Insert/edit link - creates a link from selected text. See chapter Creating links for more details. |
|  | Insert/edit image - inserts an image into the current cursor position. See chapter Inserting images for more details. |
|  | Insert BizForm - inserts an on-line form into the text. You can find more details on BizForms in chapter BizForms module overview. |
|  | Insert inline control - inserts an inline control into the text. You can find more details on inline controls in chapter Inline controls overview. |
|  | Insert poll - inserts a poll into the text. You can find more details on polls in chapter Polls module overview. |

### 6.8.2 Inserting images

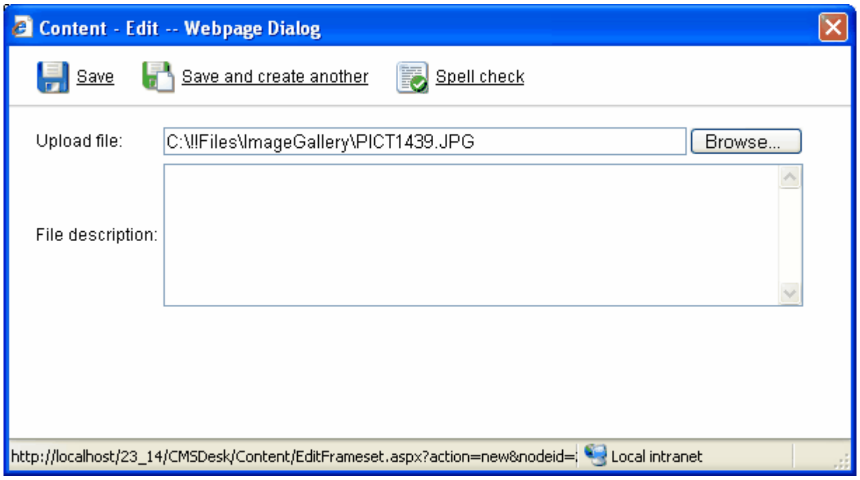
You can insert an image into the text using the Insert/edit image button in the WYSIWYG editor toolbar. The Image properties dialog opens:



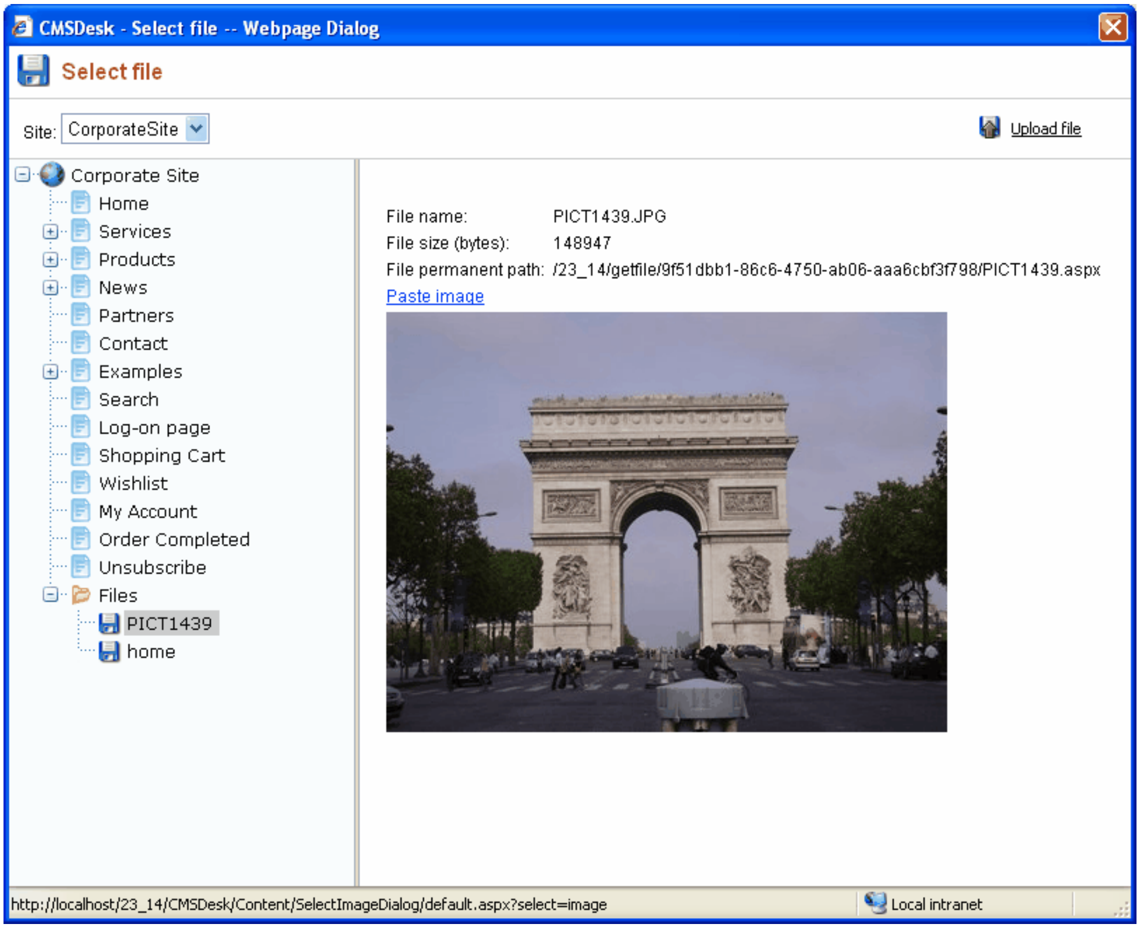
You can either insert the URL address of the image if you know it or you can click **Browse Server** and choose the image in the content tree:



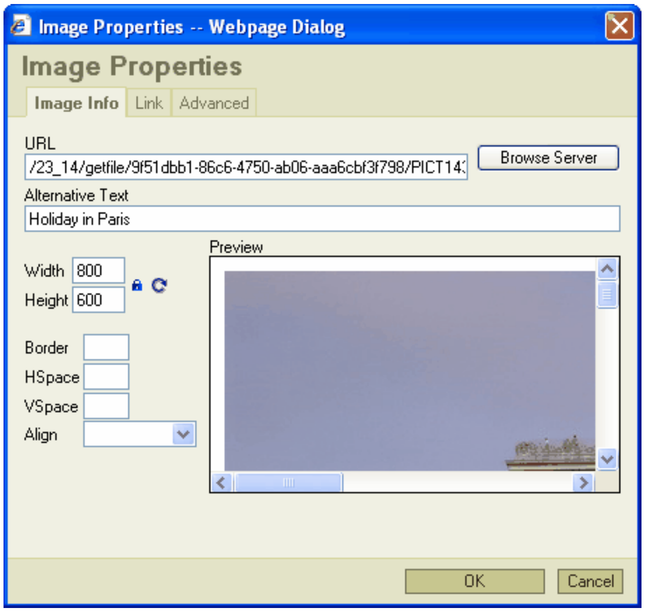
If the image hasn't been uploaded yet, you can click **Upload file** in the upper right corner and choose the file to be uploaded from your local computer:



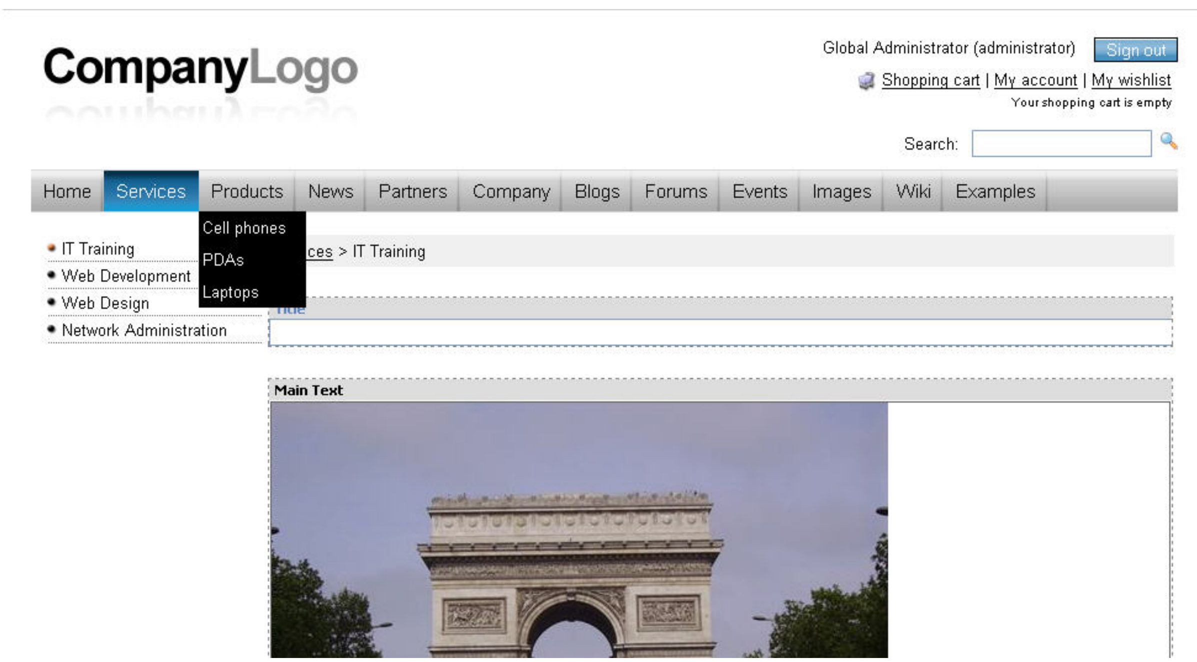
Click **Save** to upload the image and then click **Page image** in the image preview:



Now you can see the preview of the image in the Image Properties window and change its alignment, size or add alternative (ALT) text for better usability.

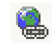



Click **OK** to insert the image into the text:

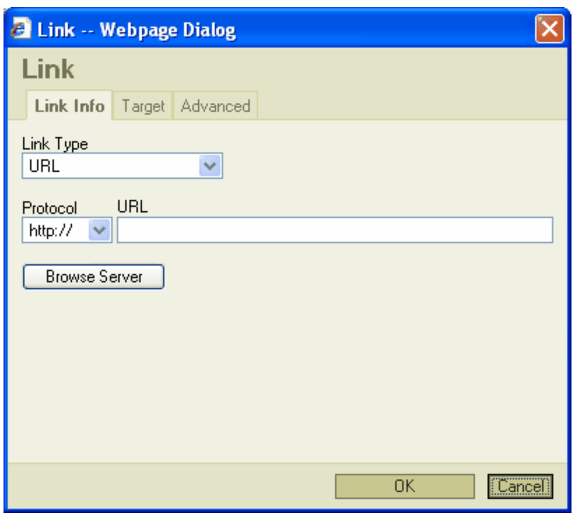



Click **Save** to save the changes.

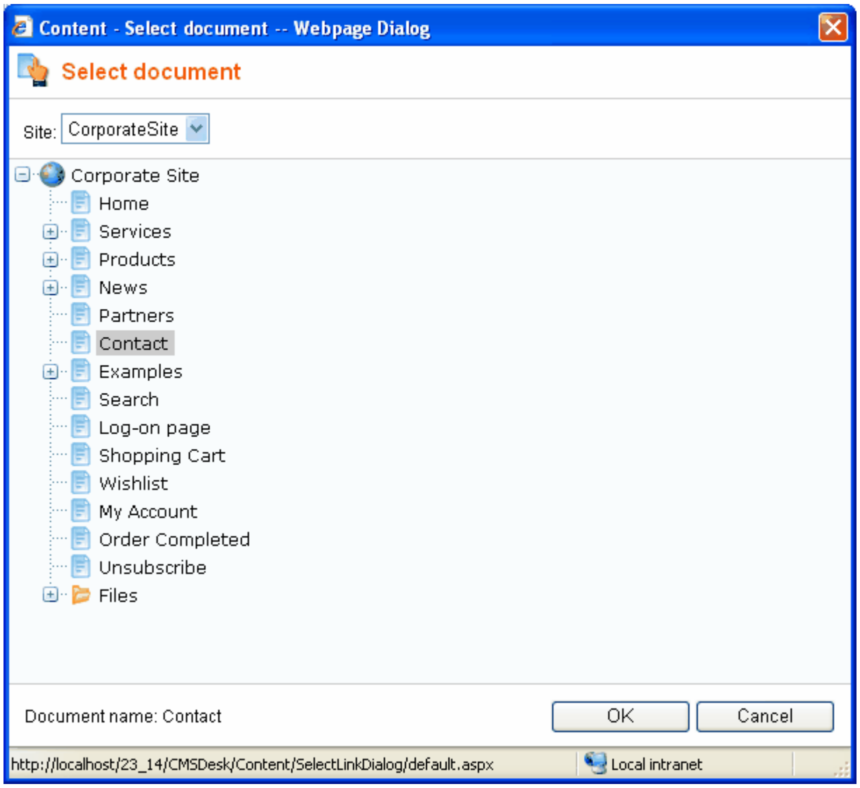
## 6.8.3 Creating links

You can create a link by selecting the text that will serve as a link and clicking the  Insert/edit link button in the editor toolbar.

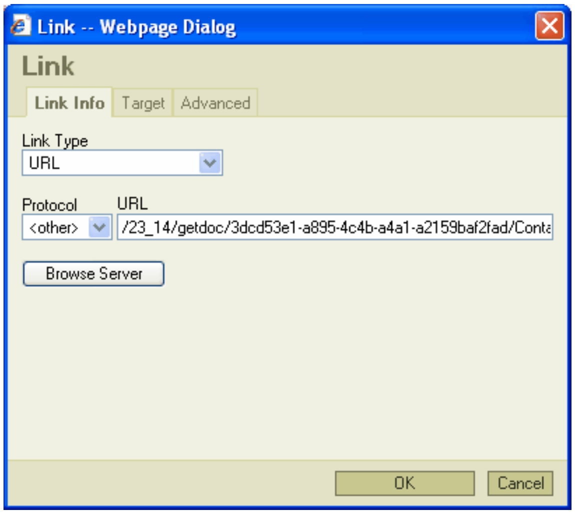

The **Link** dialog opens:



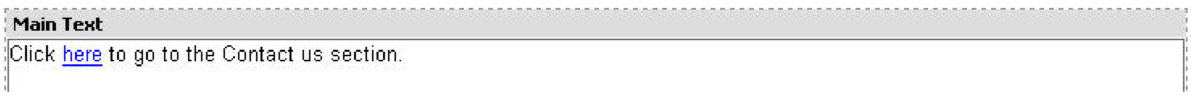
You can enter the target URL or you can click **Browse Server** and choose some of the pages of your web site:



Choose the page and click **OK**. Now you can preview the URL:




You can also change the target frame (e.g. new window) on the **Target** tab and specify link title for better accessibility on the **Advanced** tab. Click **OK** and the selected text is now used as a link to the chosen URL:




Click **Save** to save the changes.

**Link format**

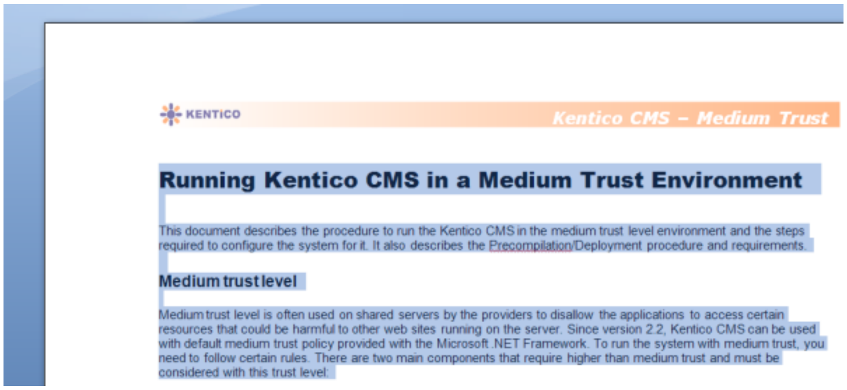
The default link format used by Kentico CMS is based on the document GUID (unique identifier). It ensures that the link keeps working even if the page is moved to some other place or its alias path is changed.

If you prefer to use a more SEO-friendly link format, you can configure it (if you're a global administrator) in **Site Manager -> Site Settings -> URLs -> Use permanent URLs**.

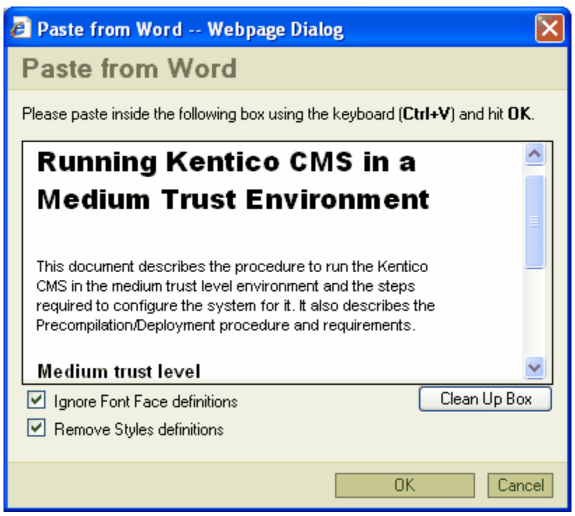
## 6.8.4 Copy & Paste from Microsoft Word

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.

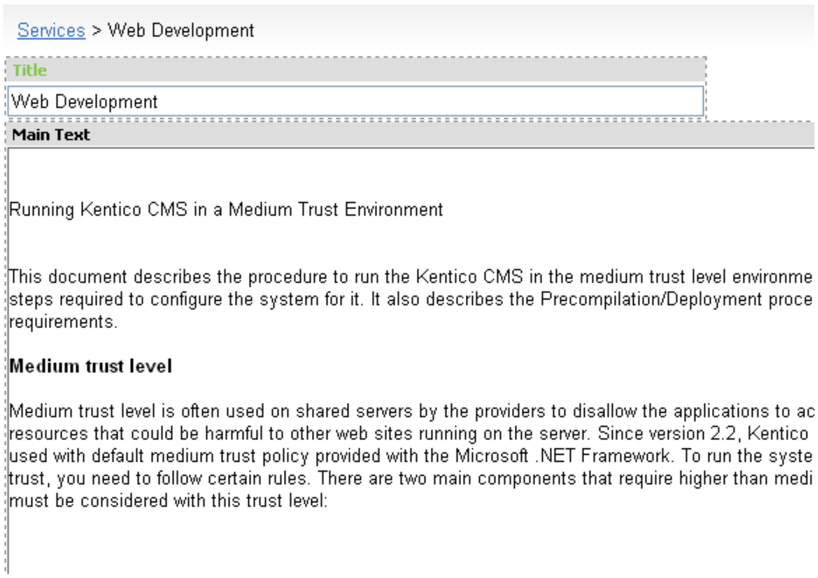
Select the text in Microsoft Word document and copy it to the clipboard (**CTRL+C**):




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 web site 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.

### 6.8.5 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 keep the web site design consistent and clean.

## Defining the toolbar sets

You can define toolbar sets in file <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["MyToolbar"] = [
       ['Source','-','Preview'],
       ['Bold','Italic','-','OrderedList','UnorderedList','-','Link','Unlink'],
       '/',
       ['Image','Flash','Table','Rule','Smiley','SpecialChar','PageBreak'],
] ;
```

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.

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 apply. 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 our example, it's MyToolbar).

Your toolbar will look like this:


## 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.

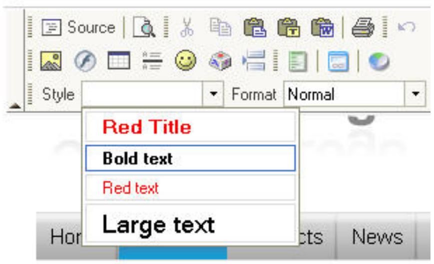
**Standard toolbar sets**

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)

## 6.8.6 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:

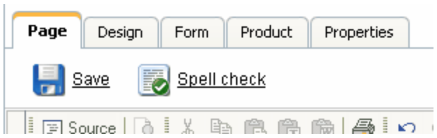
```
.GreenText { color: green; }
```

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

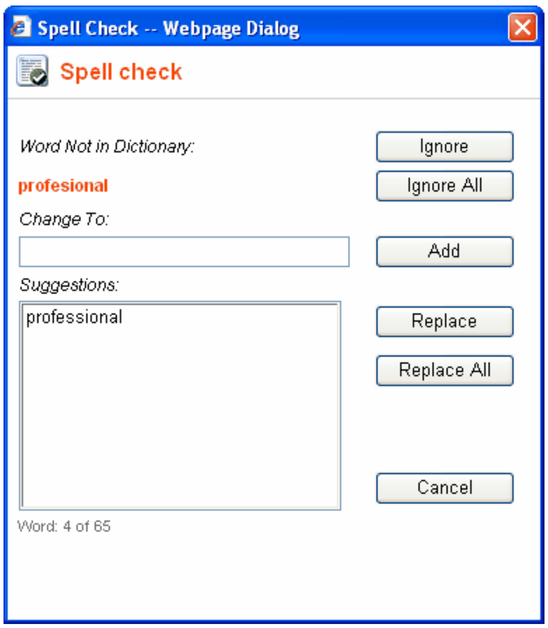
Every time you modify the fckstyles.xml file (or any other file that is used for the WYSIWYG editor), you need to clear your browser cache so that the changes apply. In Internet Explorer 7.0, you need to click **Tools -> Delete browsing history...** and click **Delete files...**

## 6.9 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>\AppData\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 web site 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.

## 6.10 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.

## 6.11 Permissions and security

Kentico CMS allows you to configure permissions for particular site sections or even particular documents. You can find more details in chapter Security and administration.

# 6.12 FAQ

This chapter provides most often 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 web site 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 web site immediately.
2. **Workflow** - if you use workflow, the changes are published on the live web site 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 VII

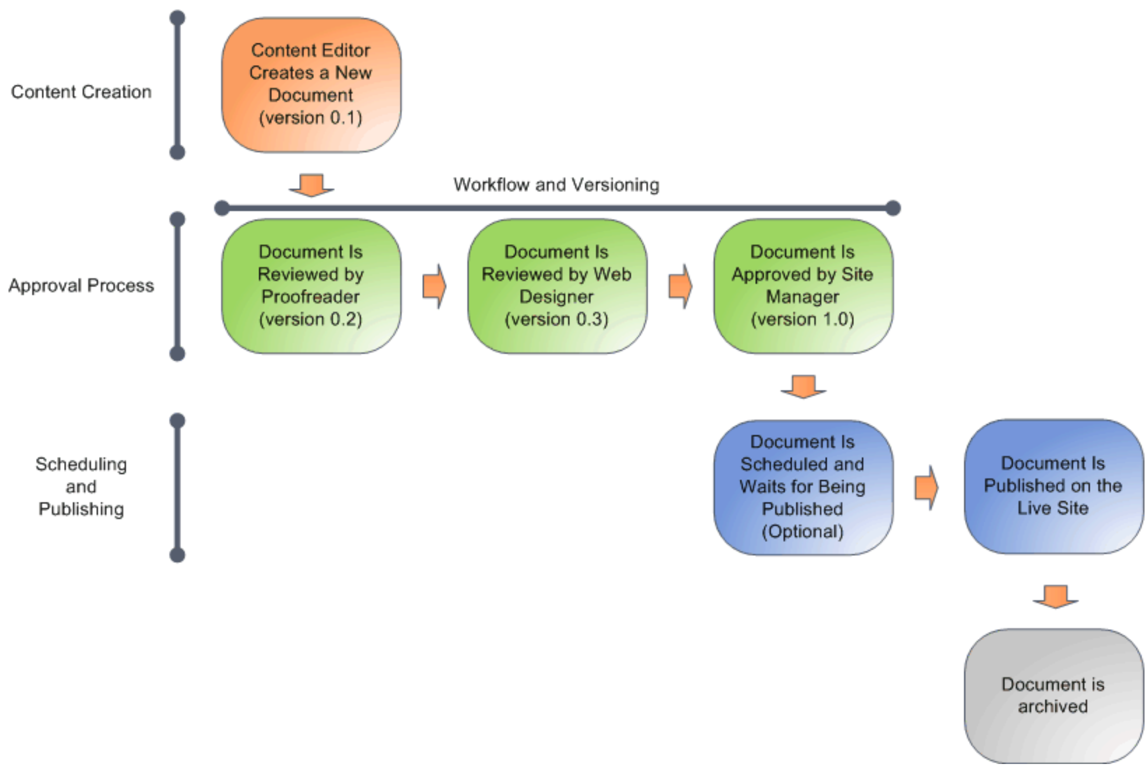
# 7 Workflow and versioning

## 7.1 Workflow 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 web site.
- The **versioning** support is tightly bound with workflow and it allows you to store (audit), view and roll-back previous versions of the content.
- 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 they are approved to the *Published* step.
- The **archiving** support allows you to archive a document. Such a document is no longer displayed on the web site, 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. For each step, the members of the authorized roles are allowed to modify and approve/reject the document. The document can be approved by any of the authorized users.**

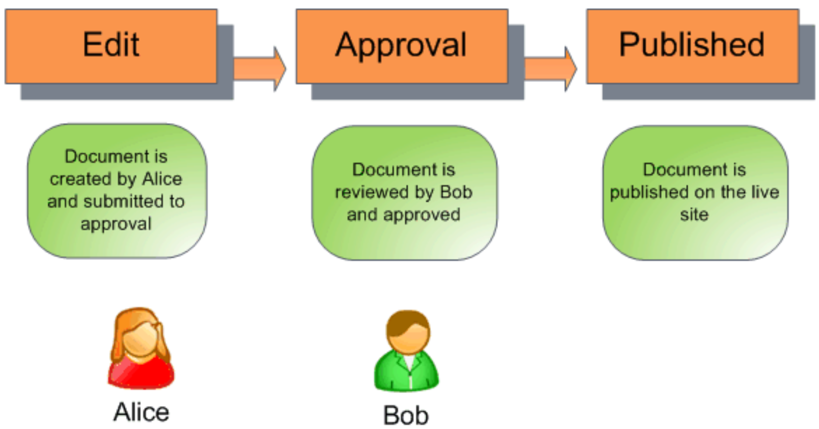
As you can see, the workflow process can be used to ensure quality of the content copy and design. In the following chapters, you will learn how to define and apply workflow processes.

## 7.2 Defining a workflow

Workflow settings can be managed in **Site Manager -> Development -> Workflows**.

The workflow you create in this chapter will be used for all examples in the next chapters. It will look like this:

- Any editor can create a news item in the /News section of the web site.
- Then, the news item must be approved by one of the members of the role PR Managers. The users authorized to approve the document in the given step are also authorized to modify the document in the given step.
- Approved news item is published on the web site.




### Creating testing users and roles

Before we configure the workflow, we will create testing users. Go to **Site Manager -> Administration -> Users** and click **New user**. Create a new user with following details:

- User name: **Alice**
- Full name: Alice Murphy
- E-mail: <your e-mail>
- Enabled: yes
- Is editor: yes

Go to the **Sites** tab and assign the user to the appropriate web site.
Go to the **Roles** tab and assign the user to the **CMS Editors** role of the given web site.

Create another user with following details:

- User name: **Bob**
- Full name: Bob Johnson
- E-mail: <your e-mail - you may want to use a different e-mail>
- Enabled: yes
- Is editor: yes

Go to the **Sites** tab and assign the user to the appropriate web site.
Go to the **Roles** tab and assign the user to the **CMS Editors** role of the given web site.

Go to **Site Manager -> Administration -> Roles**, choose the appropriate site from the top drop-downlist 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 set for the purpose of this example. This means that if a user was a member of only this particular role, she 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 probably want to go to Site Manager -> Administration -> Permissions and give the user role appropriate permissions, especially those for Module CMS Content.

See also chapter Permissions for modules and documents for detailed information on how to do this.

## Defining a new workflow

Go to **Site Manager -> Development -> Workflows**, click **New workflow** and enter the following values:

- Display name: News approval
- Code name: newsapproval

Click **OK** to create the workflow.

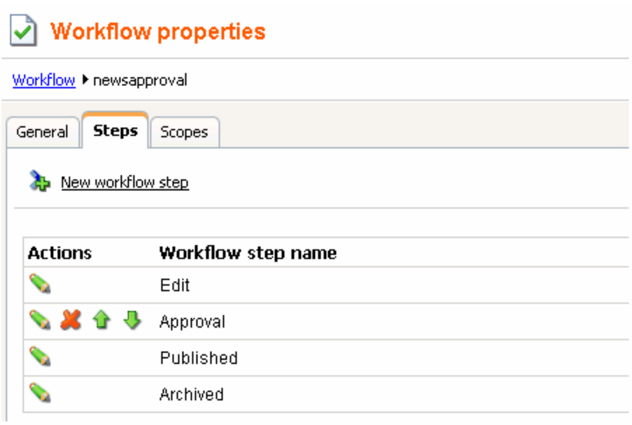
## Defining workflow steps

Choose the **Steps** tab and create a new workflow step:

- Display name: Approval
- Code name: Approval

Click **OK** and choose the **Roles** tab in the Workflow properties dialog. Choose the appropriate web site in the drop-down list and select the **PR Managers** role and click OK. Now the PR Managers are authorized to edit, approve and reject documents in the newly created step.

When you see the list of workflow steps now, it looks like this:




As you can see there are always three default steps:

- **Edit** - the first step when the document is 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 step **Edit**.
- **Archived** - the document is not published on the live site, but it remains in the content repository.

As you can see, the custom steps are placed between the **Edit** and **Published** steps. You can change the order of the steps using the up/down arrows (if there's more than one custom step).

## Defining workflow scope

The workflow scope defines which documents the workflow should be applied to. Switch to the **Scopes** tab, choose the appropriate web site from the drop-down list and click **New workflow scope**. Enter the following details:

- Starting alias path: /news (this is the section of the web site for which the workflow will be applied)
- Document type: News

Click **OK**.

We have configured the workflow process for the news documents. In the next chapter, you will learn how the workflow works in our sample scenario.

# 7.3 Using workflow

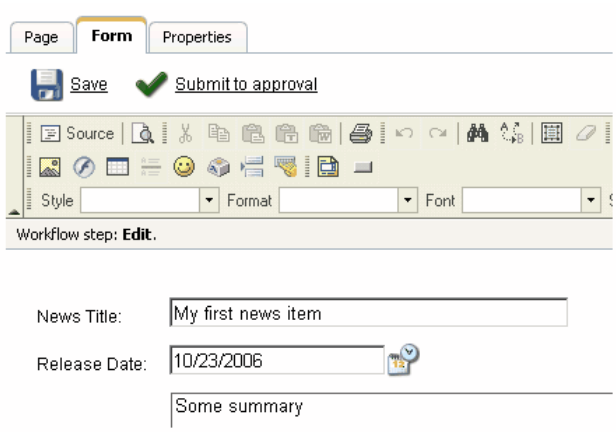
This example explains how the workflow process works from the perspective of content editors. It requires the system to be configured as described in the previous chapter.

## Creating a new document

Sign in to **CMS Desk** as **Alice** and create a news item in the /News section:

- News title: My first news item
- Release date: <use the date picker to choose 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's a new button **Submit to approval** displayed in the document menu:




Before we submit the document to approval, we will preview it. Click **Preview** in the main toolbar and see what the document will look like on the live site.

Now go back to the editing form and click the **Submit to approval** button. The document is now waiting for the 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

Sign in to **CMS Desk** as **Bob** and select the **/News/My first news item** document. You can now preview, edit and approve or reject the document. Once you approve the document, it's published on the live site. A notification e-mail is send to Alice so that she knows the document has been published.

**Please note:** When you're approving or rejecting the document, you can go to **Properties -> Workflow**, add additional comments and click **Approve**.

**Tip: Reports "Documents waiting for my approval" and "Recent documents"**

Every user can find the list of documents waiting for his/her approval in dialog **My Desk -> Documents waiting for my approval**.

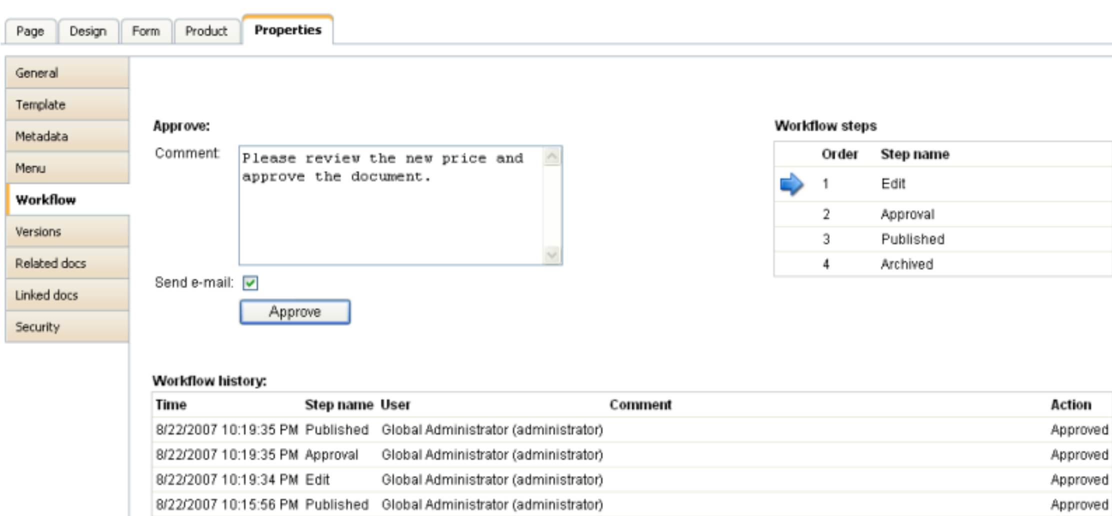
Similarly, the user who submitted the document can see the document status in dialog **My Desk -> Recent documents**.

## 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 he/she is 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**.

## 7.4 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 when the document status changes.

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

# 7.5 Versioning and rollback

Versioning allows you to store all versions of the document that uses workflow. You can view the previous versions of the document in **CMS Desk -> Content -> Properties -> Versions**.


This dialog allows you to **view, rollback and delete** the old versions. If you **rollback** some of the previous versions, it's added to the top of the version history, but it needs to be approved again so that it's published.

You can use the **Destroy history** button to clear the history and restart version numbering. The user must be granted with **Destroy** permission for this document.

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

# 7.6 Content locking

Content locking allows content editors to lock the document for editing so that the other editors cannot modify the document at the same time.

## Content locking configuration

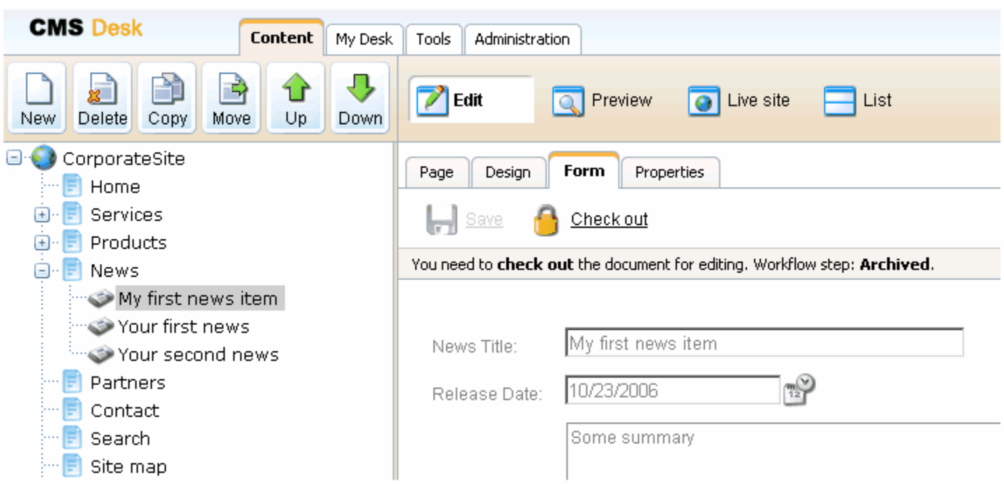
You can enable content locking for all versioned documents in **Site Manager -> Settings -> Content management -> Use check-in/check out**.

## Example

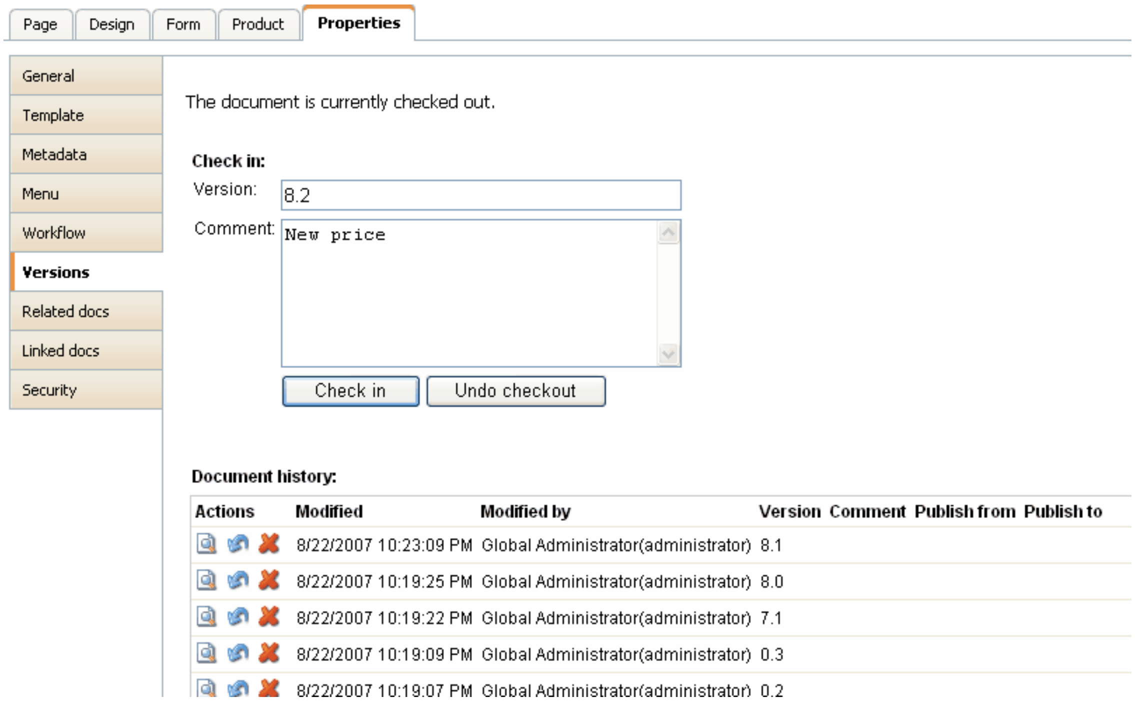
Here's an example of operations in the document life cycle:

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 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 who are granted with permission "Check-in any document" can check-in any document even if he/she hasn'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.

## 7.7 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

# 8 Security, permissions and personalization

## 8.1 Security model

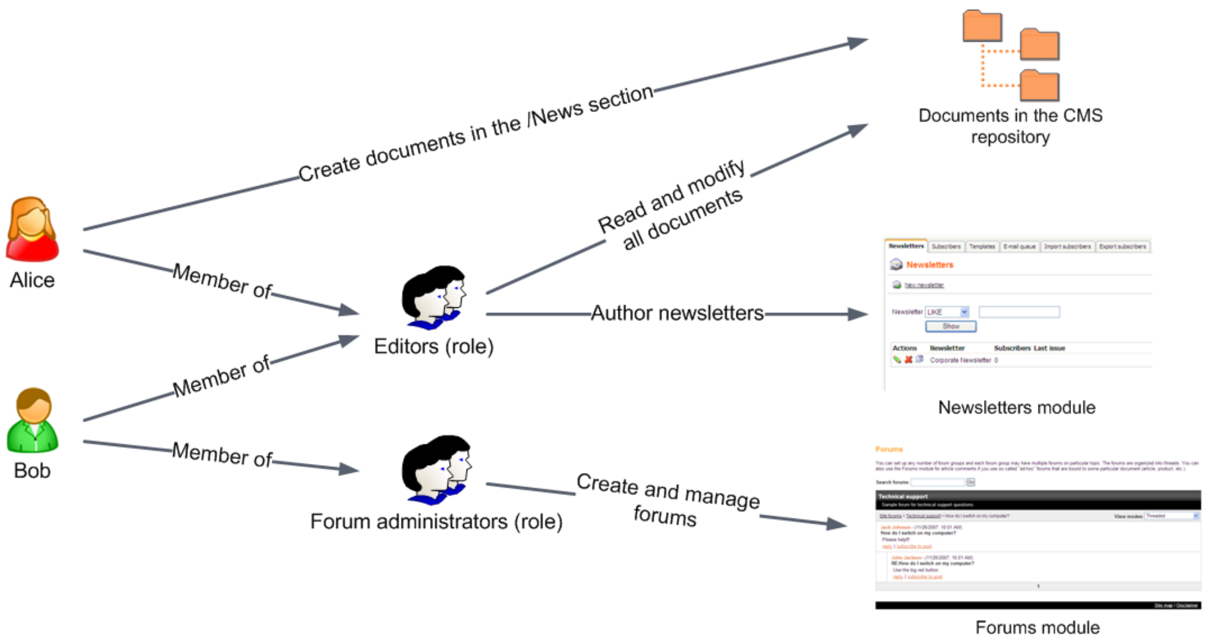
Kentico CMS provides a flexible security model that allows you to configure granular permissions to content and modules. The security model consists of:

- users (shared among web sites)
- roles (specific for web sites)
- module permissions
- document permissions

The users, roles 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 web site (the current web site is recognized by the current domain).

### Relationships between users, roles and permissions

The following figure shows how users are assigned to roles and how users and roles are granted with permissions for documents and modules:




**Users** can be member of any number of roles. They can be granted for particular documents in the CMS repository. If you want to grant a user with permissions for some module, you need to make the user a member of some role and grant the permissions to the role (the users cannot be granted with permissions for modules 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 set of permissions.

If the user is member of multiple roles, her **permissions for modules** are calculated as a sum of all permissions granted to all roles.

If the **permissions for documents** in the CMS repository are granted to both user and her roles, the document permissions are calculated as a sum of all permissions granted to the user and to all roles. If the user or some of her roles is **denied to make some action** (such as modify document), then the result is always "denied" for the given permission even if the remaining roles are allowed to make the action.

## 8.2 User management

The user can be a member of any number of roles and can be assigned to any number of web sites.

There are two important attributes of the user account:

- **Is editor** – it specifies if the user is allowed to sign in to Kentico CMS Desk.
- **Is global administrator** – it specifies if the user is allowed to sign in to Kentico CMS Site Manager and the Design tab in the CMS Desk.

**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.

The local administrators cannot modify global administrator accounts.

## Default user accounts

There are following default user accounts:

- Administrator – user with full permissions.
- Public – user that represents an anonymous visitor of the site.

### Creating a new user

Go to **Site manager -> Administration -> Users** or **CMS Desk -> Administration -> Users** and click **New user**. Now you can enter the following properties:

| | |
|---|---|
| User name | User name (login). It must be unique over all web sites. |
| Full name | User full name (first name, middle name and last name). |
| E-mail | User e-mail. |
| 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 Kentico CMS Desk. It's used to differentiate users who are only allowed to visit member areas of the web site from content editors who can use Kentico CMS Desk user interface. It provides an extra security layer. |
| Password | User password. |
| Confirm password | User password again for confirmation. |

## Editing user properties

You can edit user properties in **Site manager -> Administration -> Users** -> click the **Edit** button of the chosen the user (or in **CMS Desk -> Administration -> Users**).

**General properties**

| | |
|---|---|
| User name | User name (login). It must be unique over all web sites. |
| Full name | User full name (first name, middle name and last name). |
| First name | First name. |
| Middle name | Middle name. |
| Last name | Last name. |
| E-mail | User e-mail. |
| 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 Kentico CMS Desk. It's used to differentiate users who are only allowed to visit member areas of the web site from content editors who can use Kentico CMS Desk user interface. It provides an extra security layer. |
| Is global administrator | Indicates if the user is global administrator with full permissions for all features and data. |
| Is external user | This attribute is used when you are using an integration with an external user database. |
| Preferred content culture | Preferred culture in which the users wants to see the content. |
| Preferred user interface culture | Preferred culture in which the users wants to see the user interface (CMS Desk and Site Manager). |
| User nick name | The nick name used in Forums module. |
| User picture | The picture used in the Forums module. |
| User signature | The signature used in the Forums module. |
| Created | The date and time when the user account was created or when the user registered on the web site. |
| Last logon | The date and time when the user last logged in. |
| URL referrer | The web site from which the user came to your web site before she registered. |
| Campaign | The campaign from which the user came to your web site before she registered. See Tracking marketing campaigns for details. |
| Starting alias path | The alias path of the site section that opens in CMS Desk. If you specify this value, the user is not allowed to browse other sections of the web site 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 the given user, grant him with appropriate document permissions (Properties -> Security). |

**Password**

Here you can change the user's password:

| | |
|---|---|
| Password | User's new password. |
| Confirm password | Repeat user's new password for confirmation. |

**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 her user name and password. Simply check the appropriate boxes and click **OK** to save the changes. This dialog applies only to logging in to the CMS Desk and is intended for separation of content editors responsible for different web sites.

**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 web site) 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.

### Deleting a user

You can delete users in **Site manager -> Administration -> Users** -> click the **Delete** button of the chosen the user.

## 8.3 Role management

Roles are assigned to a particular (single) web site. 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. |

### 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. |

**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.

## 8.4 Authentication

The system supports both forms and Windows authentication. The **forms authentication** stores user names and passwords in the database and requires user to log on. The **Windows authentication** gets user identity from the network credentials and creates the user automatically in the database, including 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.

```
// 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 custom database), you need to configure the system as described in chapter Configuring custom authentication.

## 8.5 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 her 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**.
   On the **General** tab, set the following values:
   - **Is global administrator:** yes
   - **Is external user:** yes
   Click **OK**.
3. 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">
```

When using the Windows authentication, you also need to have the following settings in your web. config file so that the authentication is required for the live site:

```
<location path="">
  <system.web>
   <authorization>
     <deny users="?"/>
   </authorization>
  </system.web>
</location>
```

4. Save the modified web.config file. Close all browsers with Kentico CMS and open the web site in a new browser. Try to go to **<web project>\cmssitemanager** to make sure you are recognized as a global administrator.

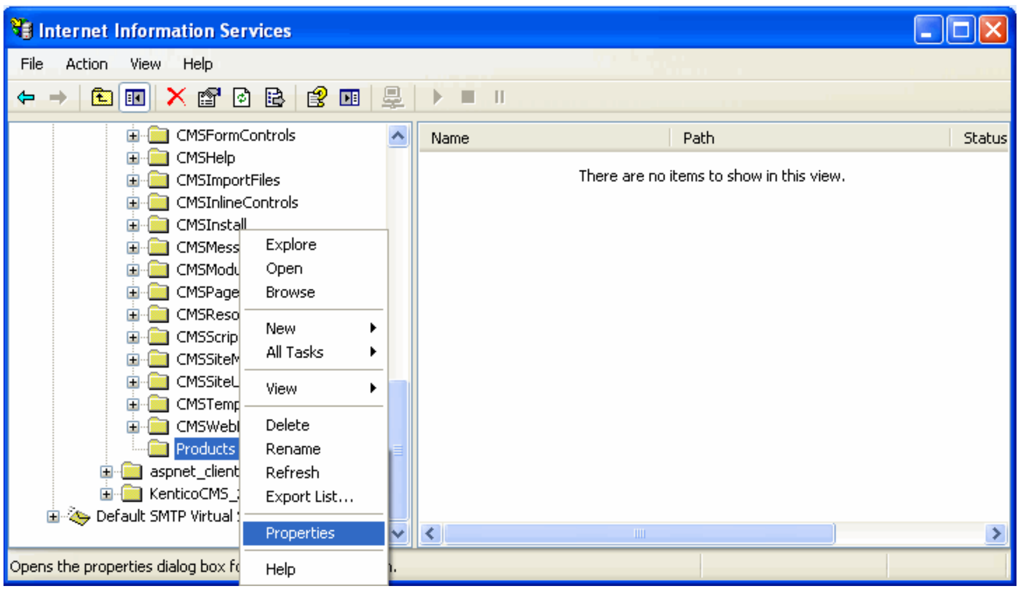
From now on, when authenticated users come to the site, their user account is created in Kentico CMS database automatically and their domain groups are imported as roles into Kentico CMS database. It means that the users are roles are not imported on some regular basis, instead they are imported when the user comes to Kentico CMS web site.

## Securing a web site section using Windows authentication

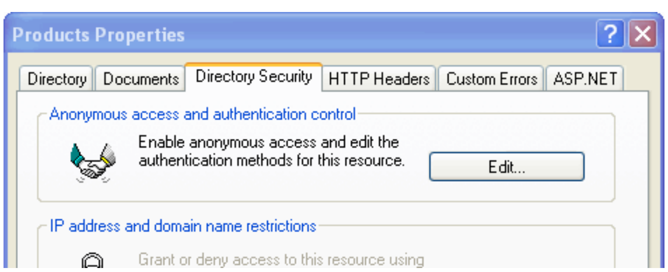
It is also possible to secure only a certain section of your website using Windows authentication. In the following example, you will learn how to set the Products section of our sample Corporate site to be secured by the Windows authentication:

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**.

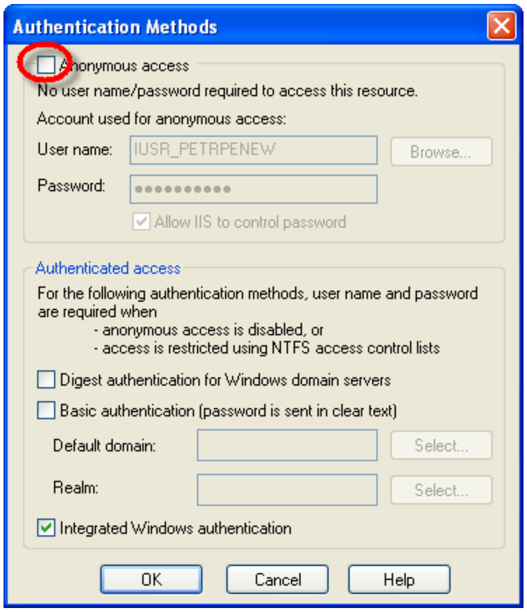
2) Open the IIS and locate the directory in the tree. Right click it and choose **Properties**.




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.

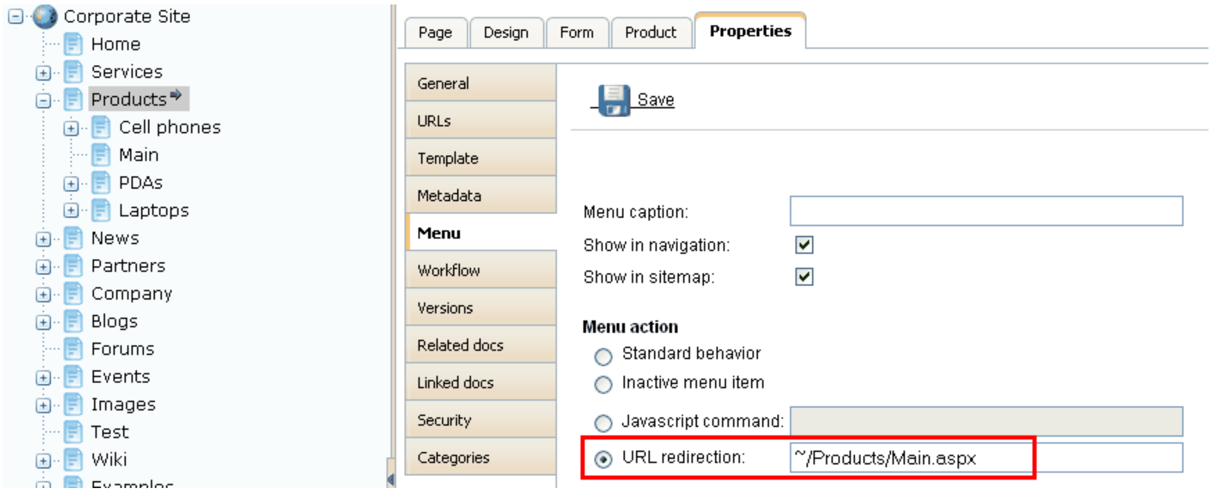



5) Open the web.config file of your web project and change value of the **mode** attribute of the **authentication** tag to **Windows**. Also find the section marked with **Windows authentication BEGIN** and change the path parameter of the location tag to the name of the created directory, which will be **Products** in our case:

```
...
<authentication mode="Windows">

...

<!-- Windows authentication BEGIN -->
       <location path="Products">
              <system.web>
                     <authorization>
                            <deny users="?"/>
                     </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.




## 8.6 Configuring mixed mode authentication

Mixed mode authentication enables signing-in using both Windows authentication and standard forms authentication at the same time. To enable this authentication mode, you have to add the following code to your *web.config* file:

```
<connectionStrings>
  <add name="CMSADConnectionString" connectionString="<LDAP connection string>"
/>
</connectionStrings>

<membership>
   <providers>
    <add name="CMSADProvider"
type="CMS.MembershipProvider.CMSADMembershipProvider"
connectionStringName="CMSADConnectionString" connectionUsername="<username>"
connectionPassword="<password>" />
   </providers>
  </membership>
  <roleManager>
   <providers>
    <add name="CMSADRoleProvider"
type="CMS.MembershipProvider.CMSADRoleProvider"
connectionStringName="CMSADConnectionString" connectionUsername="<username>"
connectionPassword="<password>" />
   </providers>
  </roleManager>
```

The LDAP connection string should be entered in the following format, where the first part is the full domain and in the second part, this domain is divided into parts using the DC parts:

```
LDAP://mydomain.mydomainxy.com/DC=mydomain,DC=mydomainxy,DC=com
```

User name and password need to be entered only on Windows XP, in the following format:

- **Username**: mydomain\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.mycompanyxy.com*). For this to work, you also have to add the following key to the AppSettings section of your *web.config* file:

```
<add key="CMSADDefaultMapUserName" value="userPrincipalName" />
```

## 8.7 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).

## 8.8 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 "/":

   **<forms name=".ASPXFORMSAUTH" path="/" ... />**

3. The web.config file of all applications uses the same machine key that is used for cookie encryption. You can find the machine key in the <system.web> section and it's defined like this:

   **<machineKey validationKey="ABCD0708...." decryptionKey="DDFF8943...." validation="SHA1" />**

4. If your applications run on different sub-domains, such as www.mywebsite.com and forums.mywebsite.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.

## 8.9 Permissions for modules and documents

The system recognizes the following permissions:

- User is a **global administrator** - she is authorized for all operations and her access cannot be denied or otherwise limited. The global administrators are the only persons who can use the Site Manager user interface.

- User is an **editor** – she can access the CMS Desk interface. This attribute doesn't implicate any particular permissions. This attribute differentiates the site editors and "registered users" who only access the front-end web site and secured site areas.

- Roles have **permissions for chosen modules** – role members can take specified actions in specified modules. These permissions are set in the **Site Manager -> Administration -> Permissions** section, in the appropriate permission matrix. You can find more details on permissions in the documentation of appropriate modules.

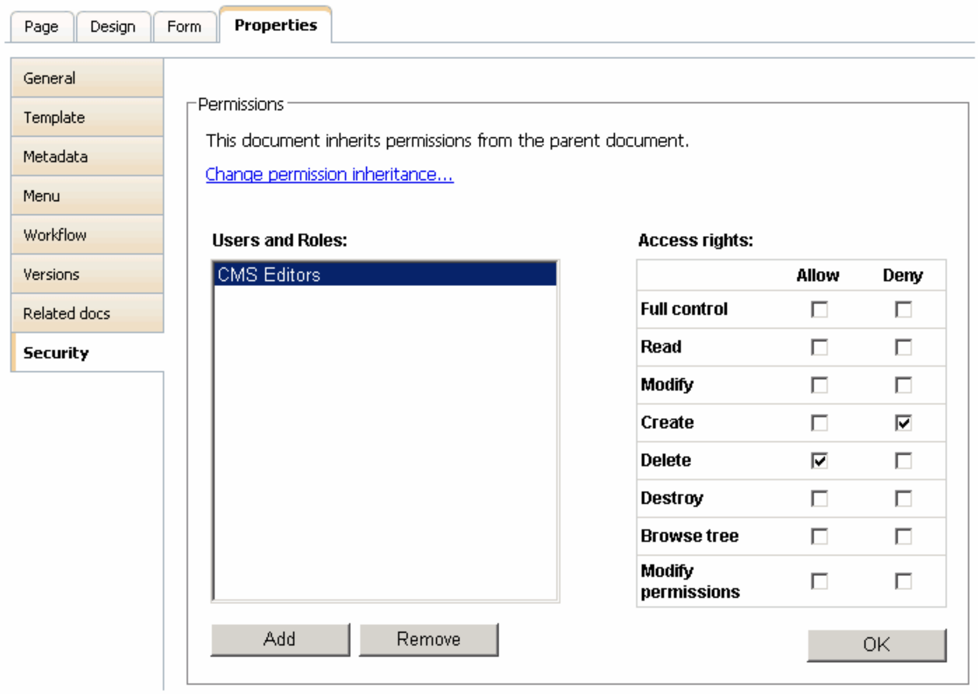
- Roles have **global permissions for all content** - these permissions can be set in the **Site Manager -> Administration -> Permissions** section, in the permission matrix called **Module: CMS Content**. Here you can set the following permissions:
  - Read - read all documents
  - Create - create any document
  - Modify - modify any document
  - Delete - delete any document
  - Destroy - destroy any document (delete without undo option)
  - Browse tree - browse content tree
  - Modify permissions - manage local permissions of any document
  - Manage workflow - approve/reject any document at any workflow step
  - Check in any document - authorizes user to check in any document even if he/she didn't check it out.
  - Design web site - this permissions allows user to edit page templates in Design mode. Please note: although the user can make the changes only for his web site, the changes may affect other web sites if he modifies a page template shared among multiple web sites.

- Roles have **global permissions for particular document type** – role members can take specified actions with particular types of documents. These permissions can be set in the Administration -> Permissions section, in the appropriate permission matrix prefixed with "Document type:".
  - 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

- Users or roles have **local permissions for particular content tree section or particular document**. These permissions are combined with global permissions for all content and global permissions for document types. If some permission is “denied” in the local permissions, the “deny” attribute overrides both local and global permissions. Local permissions are described in detail in the following paragraph.

## Managing local permissions

You can manage local permissions (i.e. permissions for particular site section or particular document) in **CMS Desk -> Content -> select some document in the content tree -> click Properties -> choose the Security tab**.




### Setting the permissions

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** button on the right. Now you can choose if the permissions should be "allowed" or "denied". You can configure the following permissions:

- **Full control** - perform all operations with 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 users with permission for the main section document, such as /products and these permissions are inherited to all child documents.

**Example**
Consider the site structure like this:

- Root
  - Home
  - News
  - Products
    - Category 1
    - Category 2

You may want to grant users with following permissions:

| | | |
|---|---|---|
| **JohnS** | **Marketing manager**<br>John can manage all content. | Grant user with Full control permission on the root or grant some of this user's roles with permissions for the CMS Content module. |
| **MarkJ** | **Product manager**<br>Mark can manage only the documents in the /Products section. | Grant user with Browse tree permission on the root so that he can browse to the Products section.<br><br>Grant user with Read, Modify, Create, Delete, Destroy and Browse tree permission on the /Products document. These permissions are inherited down to the child documents under the /Products section.<br><br>Please note that if you click on the /Products/Category 1 document, the Browse tree permission is grayed and disabled. It means this **permission is inherited** and cannot be removed - you can only **deny** the permission (unless you break inheritance - see below). |
| **AliceM** | **Copy writer**<br>Alice can modify the copy of all documents, but Mark prefers to manage the copy of the /Products section by himself only. | Grant user with Read, Modify, Create, Delete and Browse tree permission on the root.<br><br>Go to the /Products document and **deny** the Modify, Create, Delete permission for 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 **Change permission inheritance...** link in the **Security** dialog and choose one of the following options:

- **Break inheritance and copy parent permissions** - breaks inheritance and the permissions of the selected document are set to a copy of the original permissions.
- **Break inheritance and remove parent permissions** - breaks inheritance and the permissions of the selected document are cleared.

### 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**.

## Multiple Sites and Security

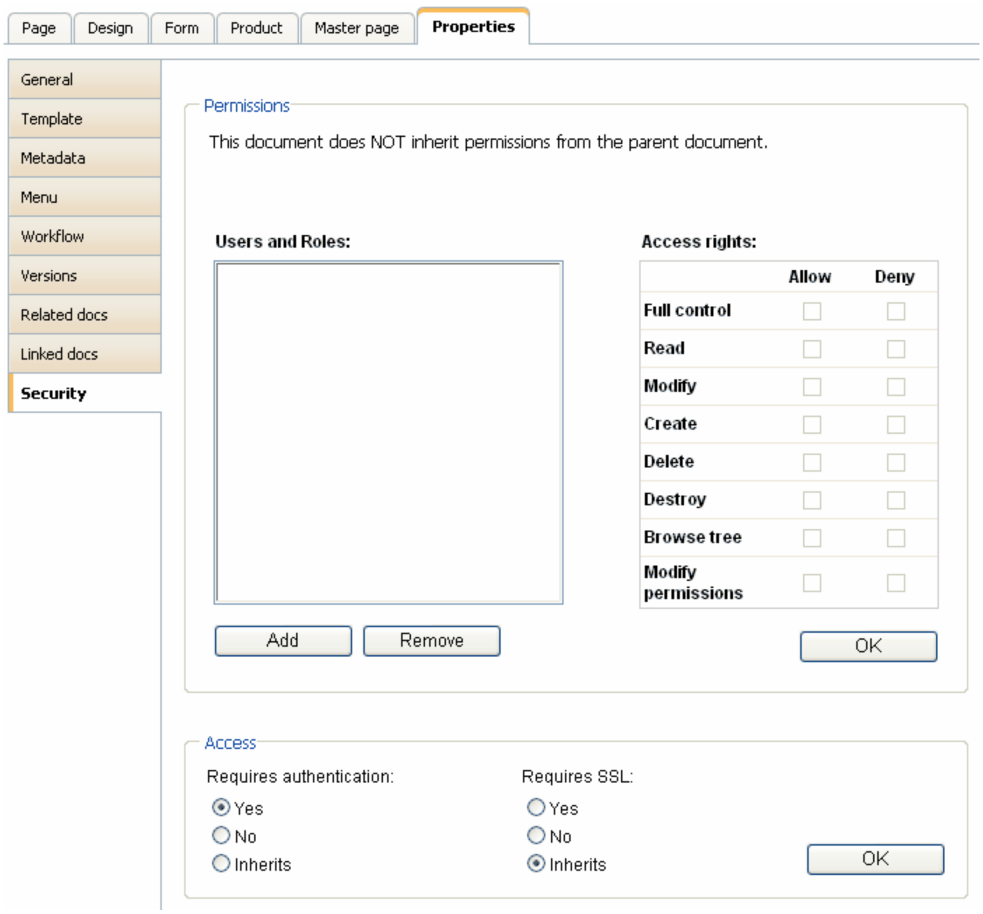
The user accounts can be shared over several web sites that are served by a single installation of Kentico CMS. You can specify which web sites an editor can edit. This applies only to the CMS Desk access. The logons to live web site (to the member sections) are not affected by assigned web sites.

Every web site has its own set of roles (there are no shared or global roles). The global permissions are assigned to these roles, which means every web site can use a different configuration of role permissions.

# 8.10 Secured web site areas

Kentico CMS allows you to easily create secured web site areas that are accessible only by authenticated users. When an non-authenticated (public) user comes to the secured section, she is redirected to the logon form.

You can mark any section of the web site as a secured site area by setting the **Requires authentication** attribute on the **Properties -> Security dialog** to **Yes**. The value is automatically applied to all underlying pages, unless they have the attribute set to **No**.




## Configuration of a secured site area

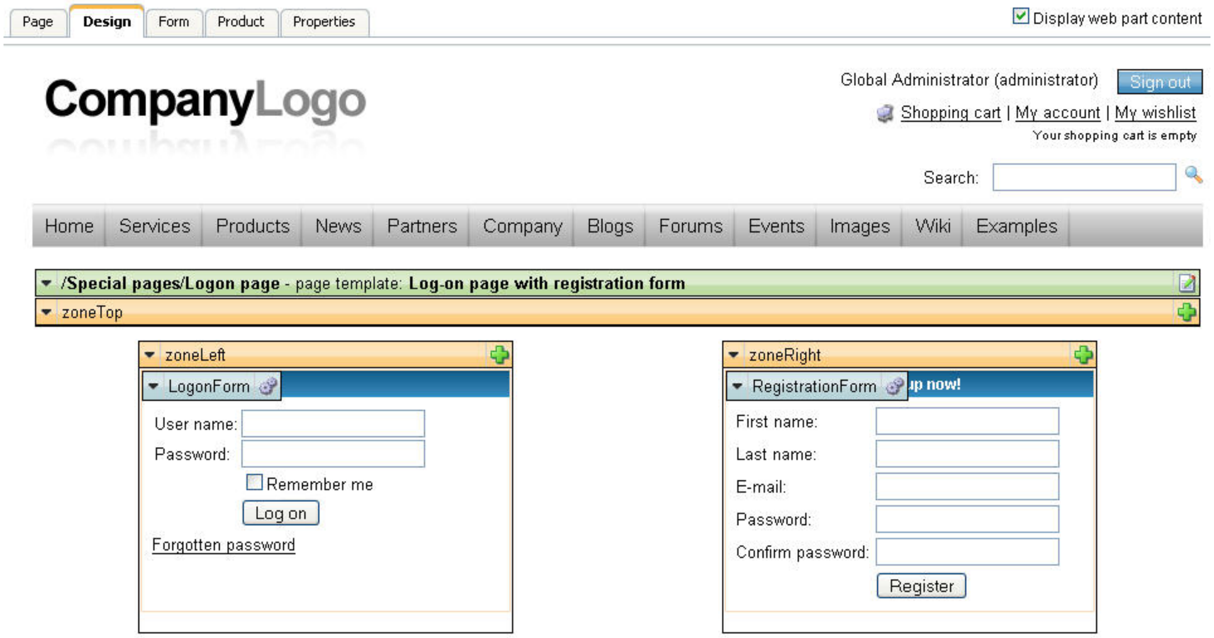
This example explains how to secure the Products section in the sample Corporate Site.

1. Sign in as administrator to **CMS Desk**. Go to the **Content** section and click the **Products** document in the content tree.

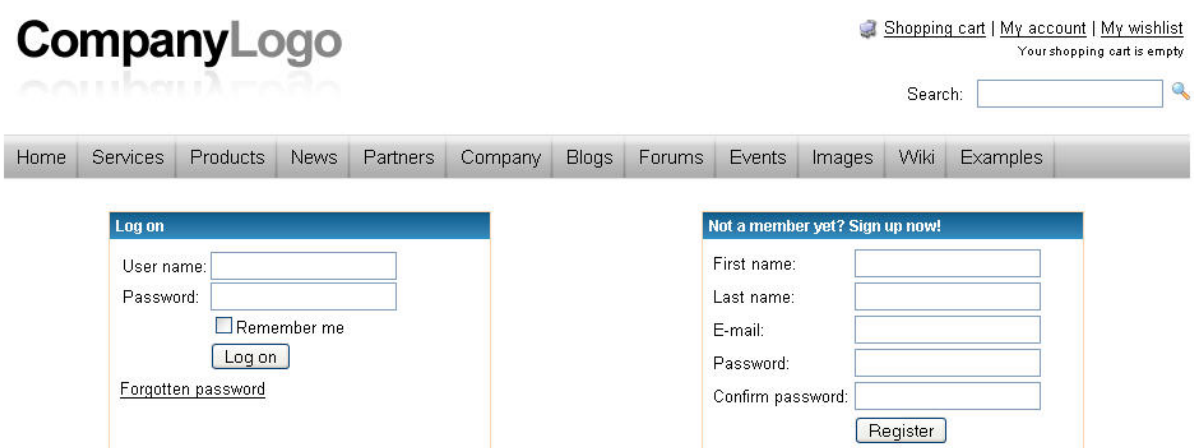
2. Click **Properties -> Security**. Set the value of the **Requires authentication** attribute to **Yes** and click **OK**.

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 for all pages within a secured area.

When the user is not authorized to read the page, she is displayed with Access denied page. You can configure custom access denied page URL in the **Site Manager -> Settings -> Security -> Access denied page URL** property.

## 8.11 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 the 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 **Anonymous User (public)**.

### Example: Personalizing the Products menu

In the previous chapter, we have configured secured area for the Products section of the web site. 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 **Roles** tab in the **User properties** dialog and add the user to the **Customers** role.

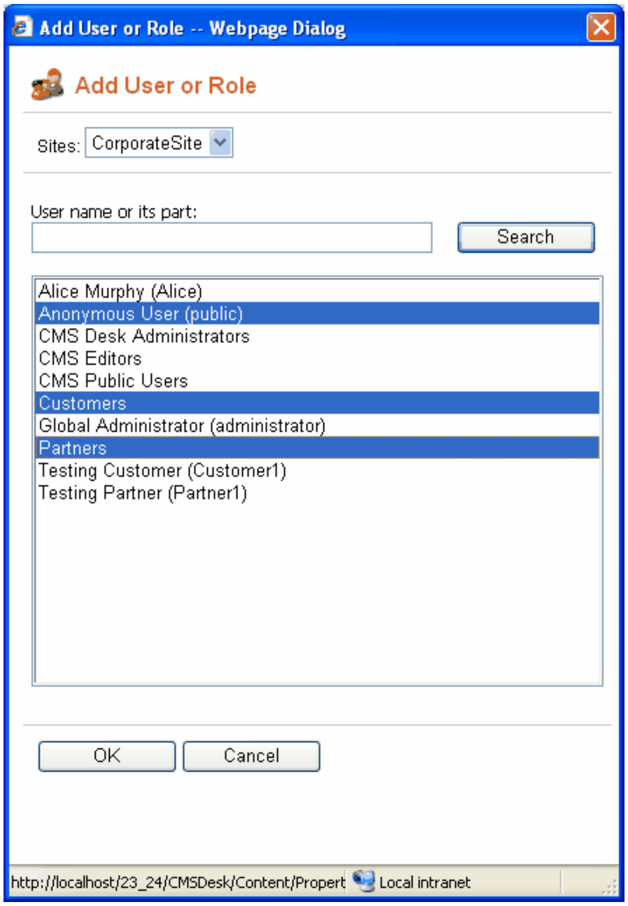
4. Create another user **Partner1** with following values:
   - User name: Partner1
   - Full name: Testing Partner

- Enabled: true
- Is editor: no

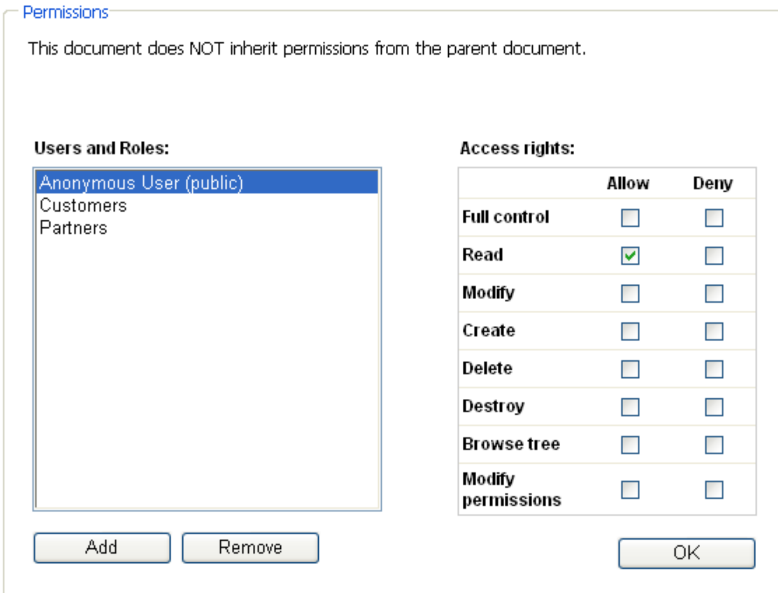
Click **OK**.
Go to the **Roles** tab in the **User properties** dialog and add the user to the **Partners** role.

5. Go to **Content** and click the root. Click **Properties -> Security**.

6. First, we need to grant the **Anonymous user (public)** and **Customers and Partners** roles with Read permission for the whole web site. Click **Add**, click **Search**, select **Anonymous user (public), Customers and Partners** and click **OK**.



7. Now grant all newly added users/roles with Read permission. You need to click each user/role, check the box next to **Read** and click **OK**.



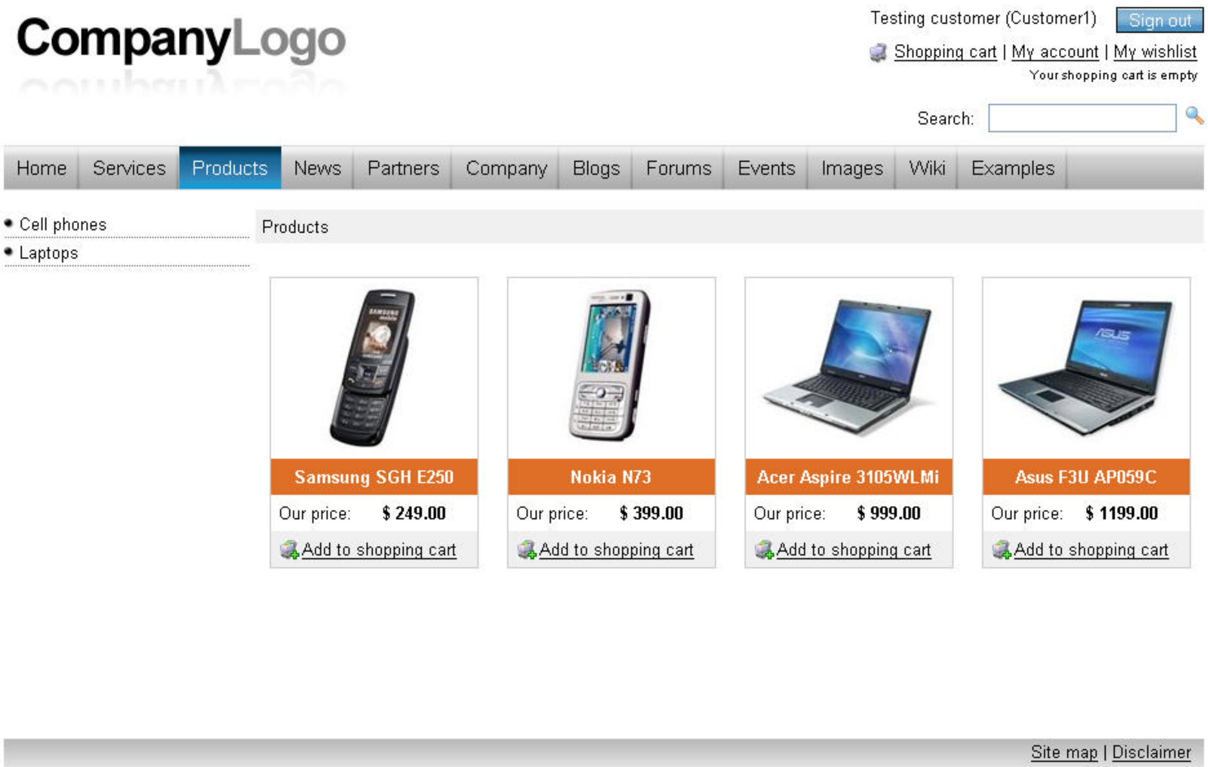

8. Now we hide the product categories from public users. In the content tree, click **/Products/Cell Phones**. Click **Properties -> Security**, choose the **Anonymous User** and **deny the Read permission**. Click **OK**. Repeat the same for categories **Laptops** and **PDAs**.

9. Now we will hide the PDAs category from the **Customers** role members. Display the **Security** dialog for the **/Products/PDAs** document and **deny the Read permission** to the **Customers** role. Click **OK**.

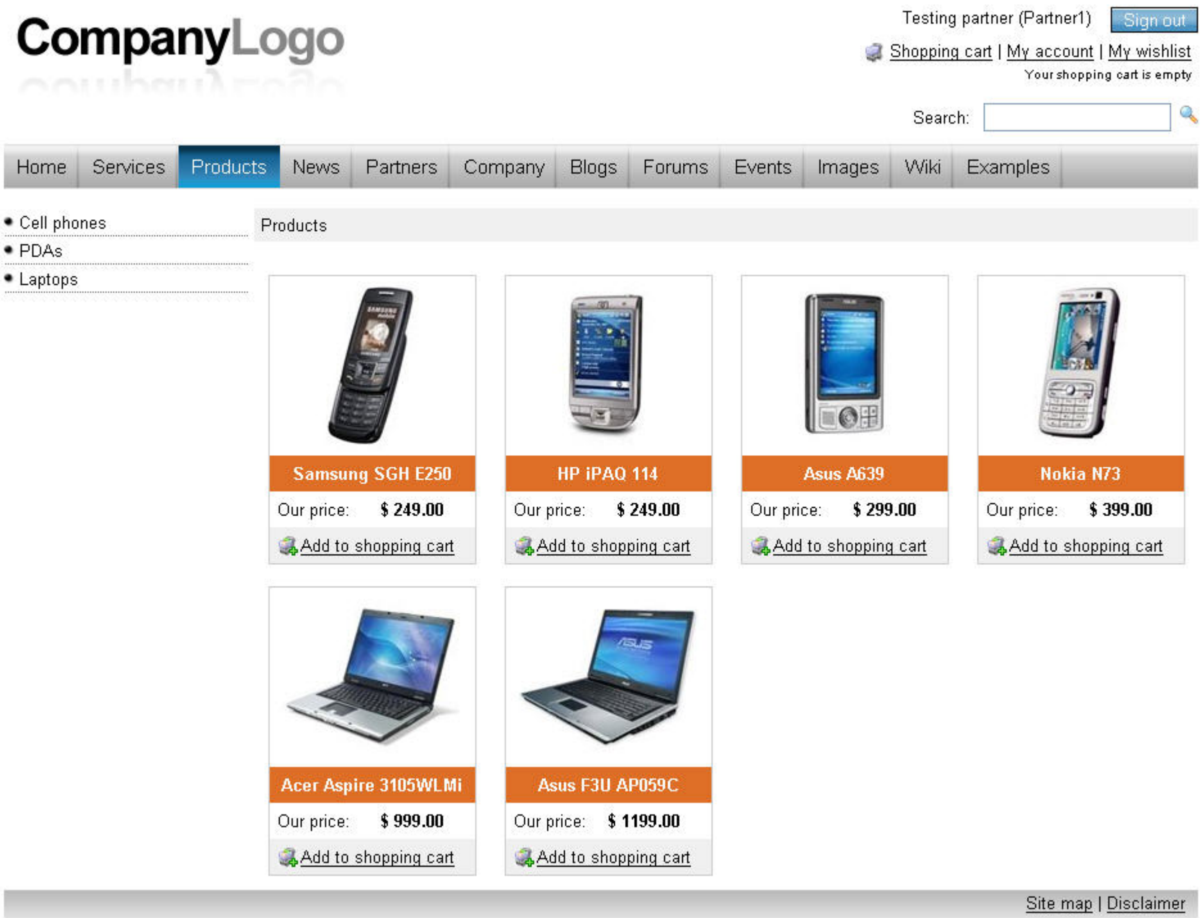
10. The permissions are not checked by web parts by default, so we need to configure the web parts so that they check the Read permission of the current user. Choose the root in the content tree and click the **Design** tab. Configure the **cmsmenu** web part and set its property **Check permissions** to true (checked). Click **OK**. Repeat the same for the **LeftTreeMenu** and **ProductDataList** web parts on the **/Products** page.

11. Sign out. If you mouse-over the **Products** menu item, you will see that the sub-categories are no longer displayed.

12.Click **Products** and sign in as user **Customer1**. 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.



13.Now sign out and sign in again as user **Partner1**.You will all categories and products:



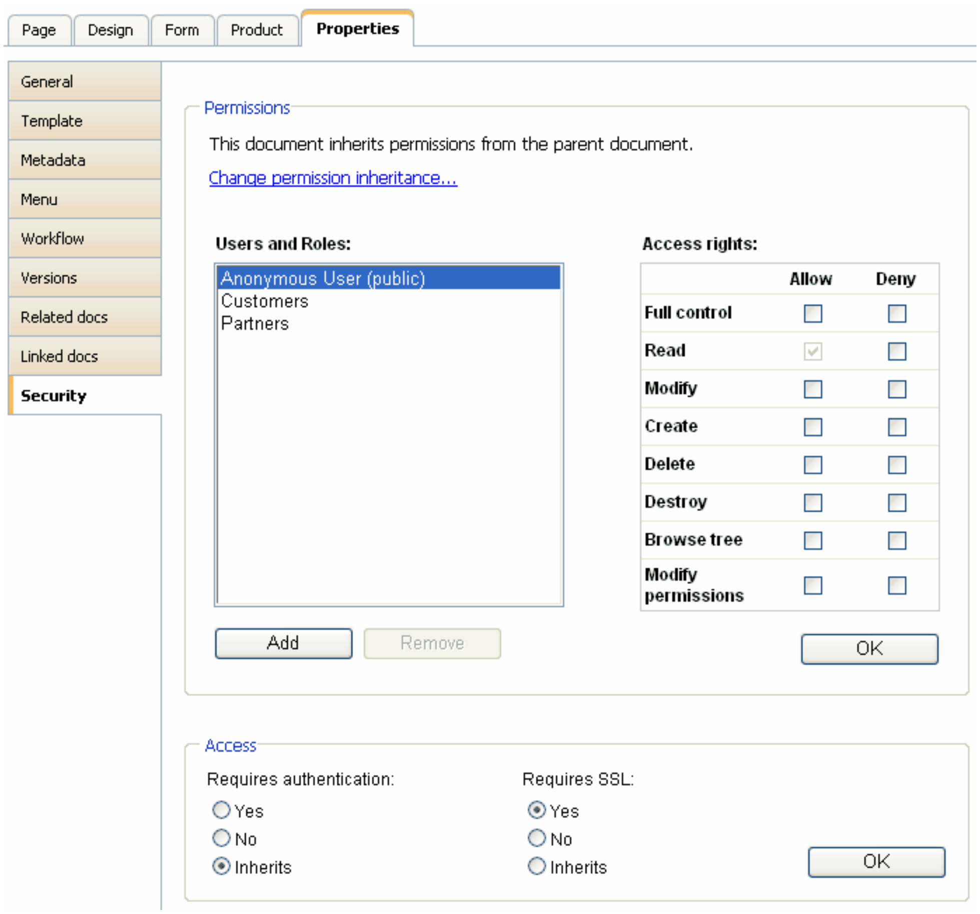

You have learned how to personalize the content displayed to users based on their permissions.

## 8.12 SSL (HTTPS) support

Kentico CMS allows you to specify which pages of the web site can be displayed only over the secured SSL (HTTPS) protocol. When a user tries to open such a page with standard HTTP protocol, she is redirected to the secured (HTTPS) URL of the same page.

**Please note:** Kentico CMS doesn't configure your web site 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**, select the document in the content tree, click **Properties -> Security**, choose **Yes** in the **Requires SSL** section and click **OK**.




## 8.13 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.

## 8.14 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 web site 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 web site, 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 web site:

```
<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.

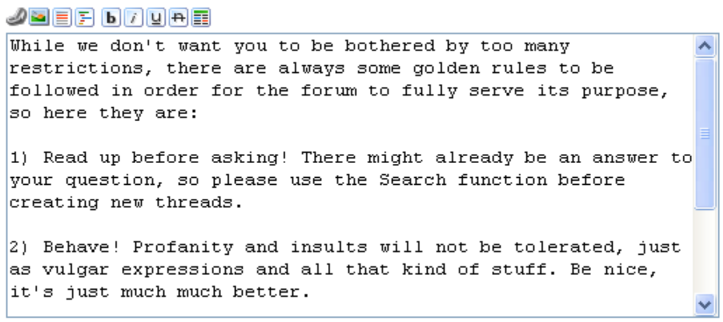
## Default allowed parameters

The common parameters of ASP.NET web forms and URL parameters **aliaspath** and **lang** are allowed by default.

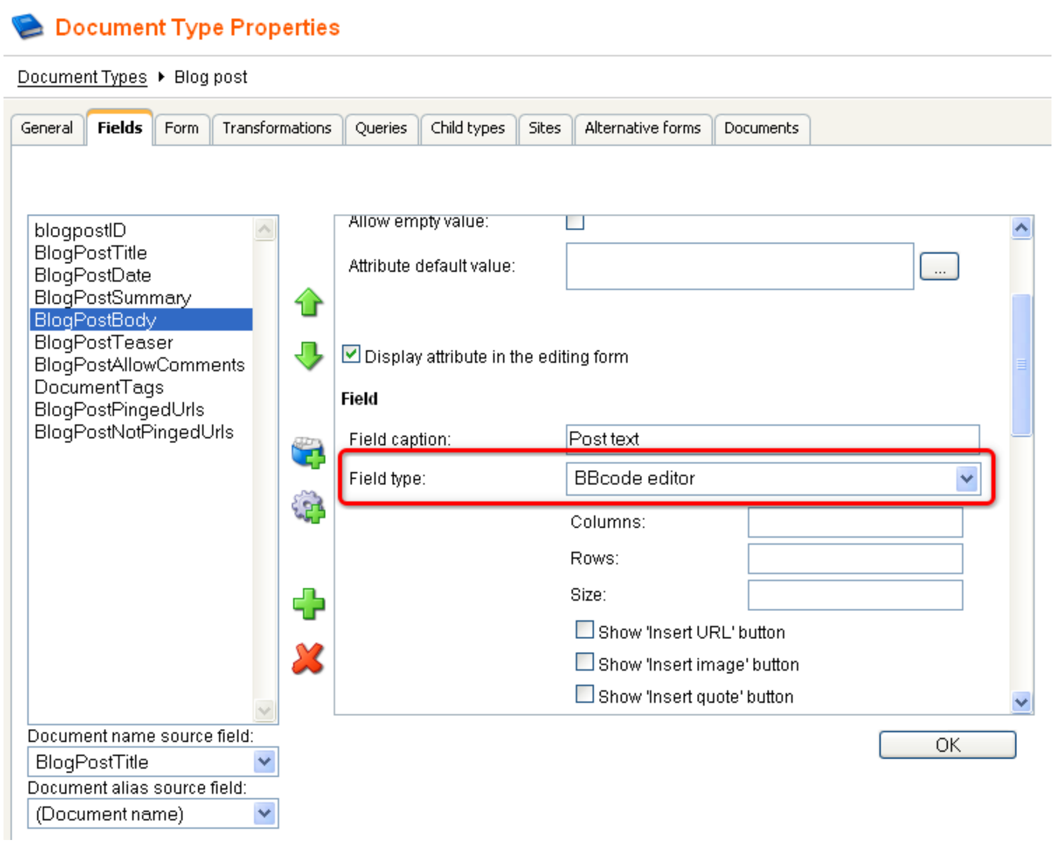
# 8.15 Cross site scripting (XSS)

For your web sites 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.:

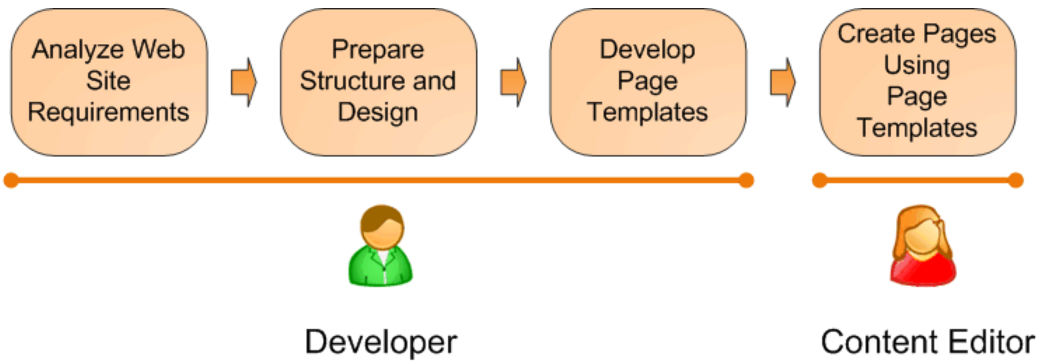
```
<%# Eval("PostSubject",true) %>
```

# Part IX

# 9 Web development overview

## 9.1 The role of web developer

The role of web developer is described on the following figure:




This figure shows how you develop the web site 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.

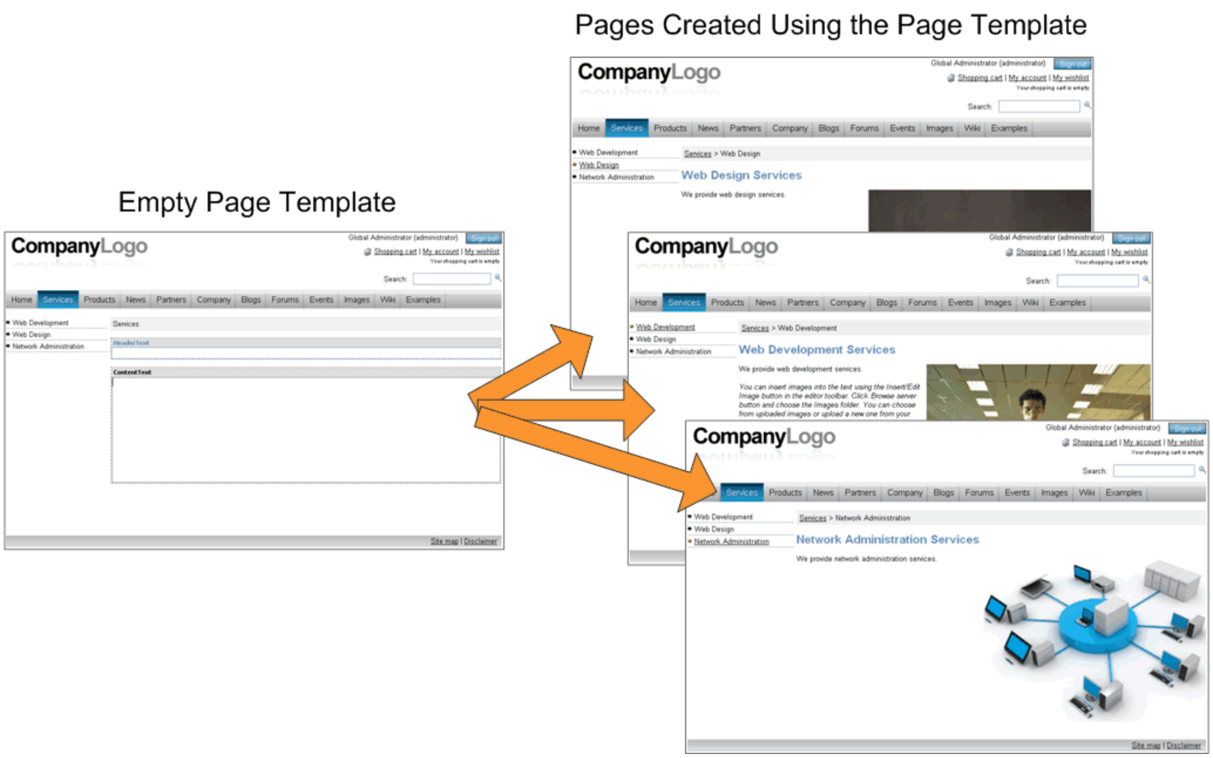
3. The developer creates **page templates** for every type of the page (home page, solutions, products, news, etc.)

4. The content editor creates new pages - she enters text and images into the page templates defined by developer.

## 9.2 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 on content editing, without taking care of the page formatting. They also help keep the web design consistent throughout the web site. The following figure shows how a single page template can be used for multiple pages:




The next chapter describes two development models of creating such page templates.

## 9.3 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 web site using a portal engine. It's the recommended way for most developers since it doesn't require programming in Visual Studio 2005. Instead, you can simply build web site using web parts in the **browser-based** user interface.
- **ASPX Templates** - this model can be chosen by advanced ASP.NET developers who prefer to create the web site using standard ASP.NET architecture and using standard development tools, such as **Visual Studio**. This model requires that you are familiar with ASP.NET development and have at least basic programming knowledge of C# or VB.NET.

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

The following table compares portal engine and ASPX templates:

| | Portal Engine | ASPX Template |
|---|---|---|
| **How you work** | You build web site using the browser-based interface.<br><br>No programming knowledge is required for common tasks. | You build ASPX pages in Visual Studio 2005.<br><br>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 web site project that you can open in Visual Studio 2005.<br><br>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 be optionally 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.<br><br>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.<br><br>You can add any custom controls and code to the web parts or user controls that you use on your web site. | You build standard ASPX pages with codebehind which means you can put any custom controls and code to the page in Visual Studio as you normally do. |
| **Advantages** | • Easier and faster to build a web site.<br>• ASP.NET programming knowledge is not required for common tasks.<br>• You can build the whole web site very quickly, only using the web browser. | • Standard ASP.NET architecture.<br>• You can use your favorite development tools, such as Visual Studio 2005 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 a full control over:

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

Also, it's important to explain that Kentico CMS is a **content management system**, not only a portal engine. It 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 interface, but you get everything from a single source, with a complete documentation and support.

# Part X

# 10 Development with Portal engine

## 10.1 Portal engine overview

Every page in Kentico CMS web sites uses some page template. **Page template** consists of 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.

The design of the page is defined by **page layout**, which is a standard HTML code. The HTML code is used to define two-column, three-column or any custom layout you can think of.

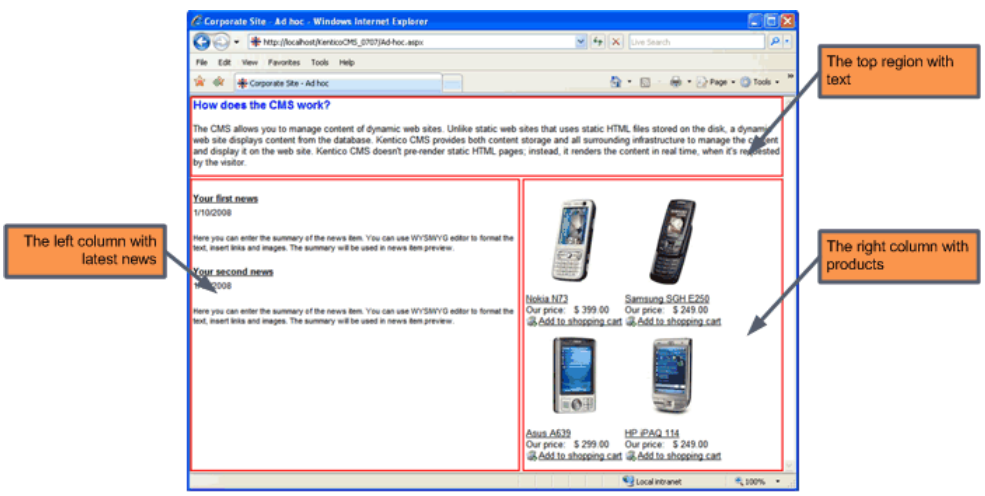
The page layout contains special markup tags that specify places where the developer can place web parts - so called **web part zones**. A web part zone can contain any number of web parts.

A **web part** (also called "servlet", "portlet" or "module" in some other solutions) is a piece of code that displays some content, e.g. a single document. Technically, web parts are standard ASCX user controls with some predefined programming interface.

The following figures show an example of simple page, its page template, layout, web part zones and web parts:

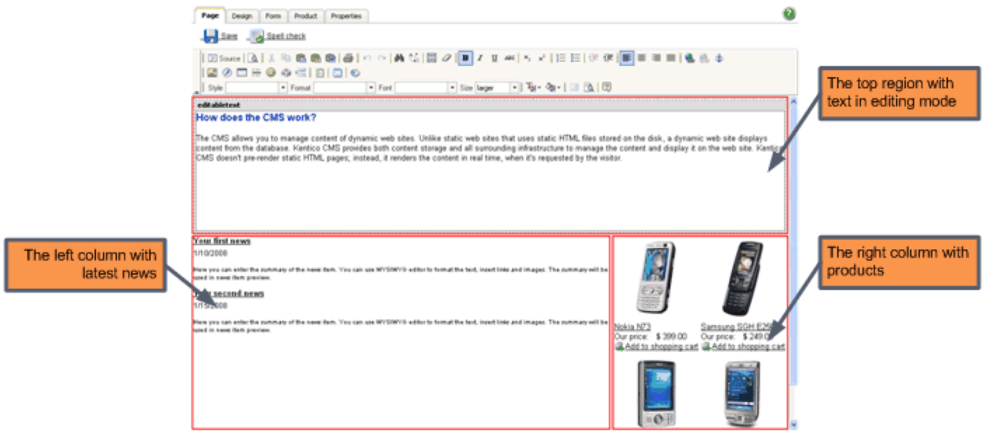
### 1) The sample page on the live site

This image shows our sample page on the live site. It displays the text on the top, news in the left column and products on 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.

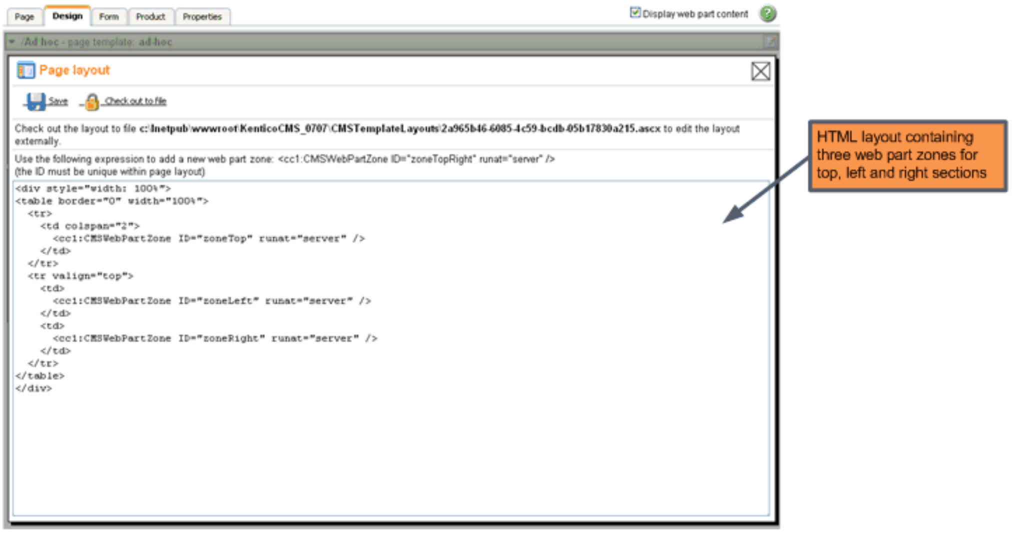



## 3) The page template of the sample page in design mode

This the design mode 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 the 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 layout here, but you can easily use CSS-based layout - you have full control over the HTML code and CSS styles.




## 10.2 Creating a new page template

There are two ways how to create a new page template:

1. Creating a blank page with ad-hoc template.
2. Cloning an existing page template.

We will start with the first option.

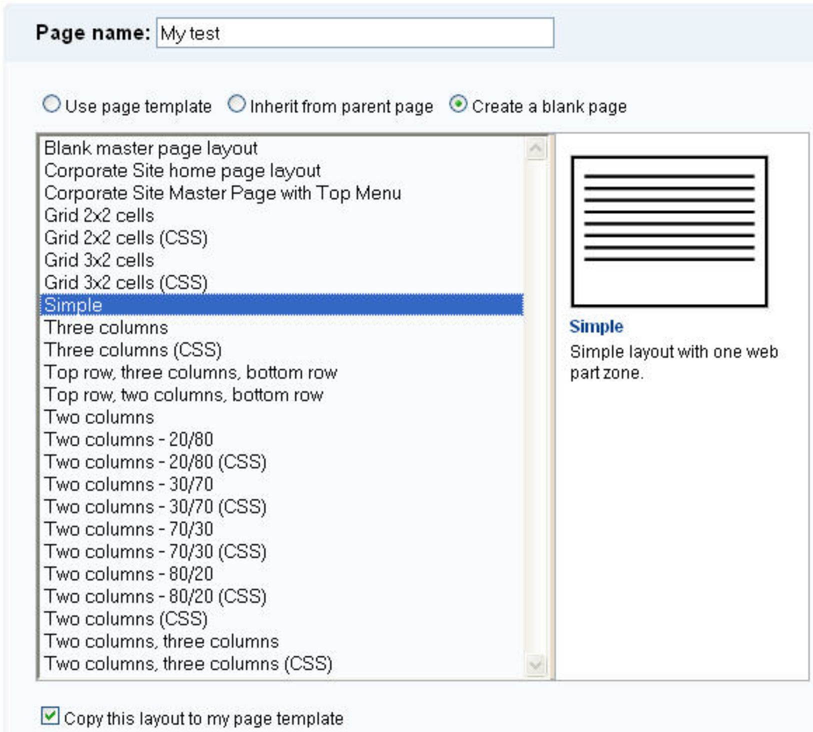
**What is an ad-hoc template?**

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 with ad-hoc template

**Please note:** It's recommended that you use the **Corporate Site** sample web site for this example.

Go to **Kentico CMS Desk -> Content**, click **New** and choose to create a new **Page (menu item)** under the web site root. Enter **My test** into the **Page name** field and choose the **Create a blank page using this layout** option. Choose **Simple** from the list of layouts and leave the box **Copy this layout to my page template** checked.

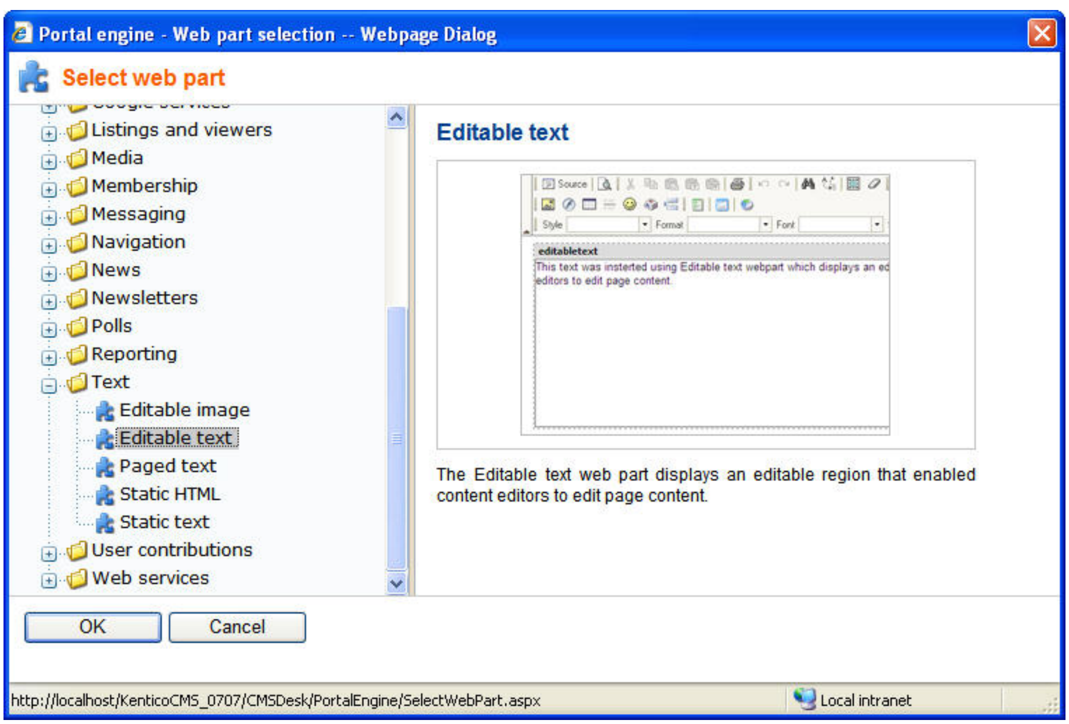


Click **Save**. The page is now displayed in the tree view and selected. Click the **Design** tab. You will see a page like this:

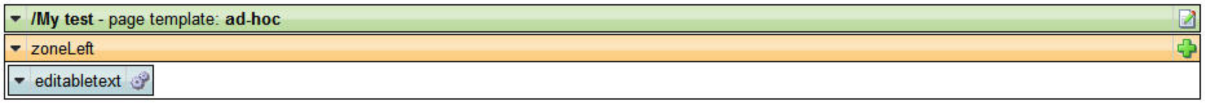

The green box with **/My test** title displays the current page. As you can see, we are now using an ad-hoc template that is specific for this page.

The yellow box with **zoneLeft** title displays the web part zone, where web parts can be placed.

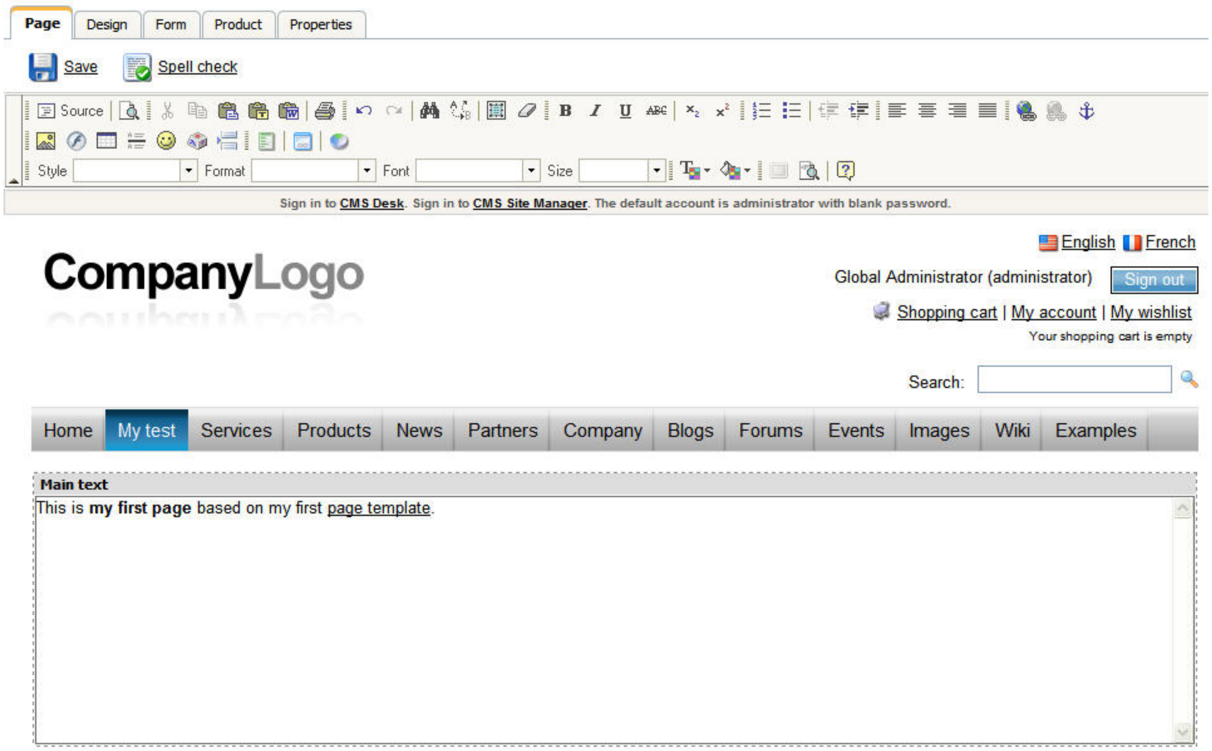
Click **+ (Add web part**. The **Select web part** dialog opens. Select the **Text/Editable text** web part and click OK. This web part allows you to create editable regions on the page where content editors can put content.



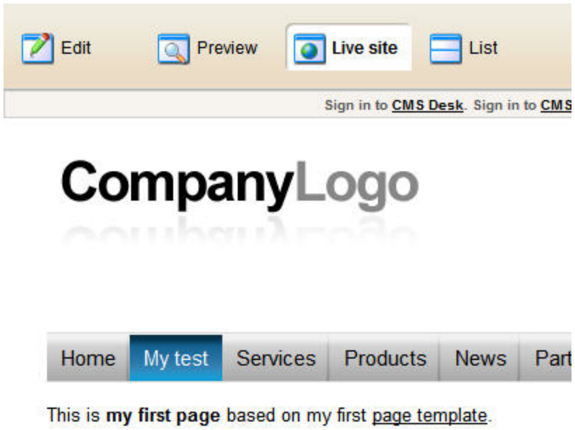
The **Web part properties** dialog opens. Set the **Editable region title** to *Main text.* Click **OK**. The web part is now added to the web part zone:




Now click the **Page** tab. You will see the page in editing mode. The page now contains your web part that displays the editable region. Enter some text into the region and click **Save**.



Click **Live site** in the main toolbar. You will see the live version of the page that is displayed to the visitors:



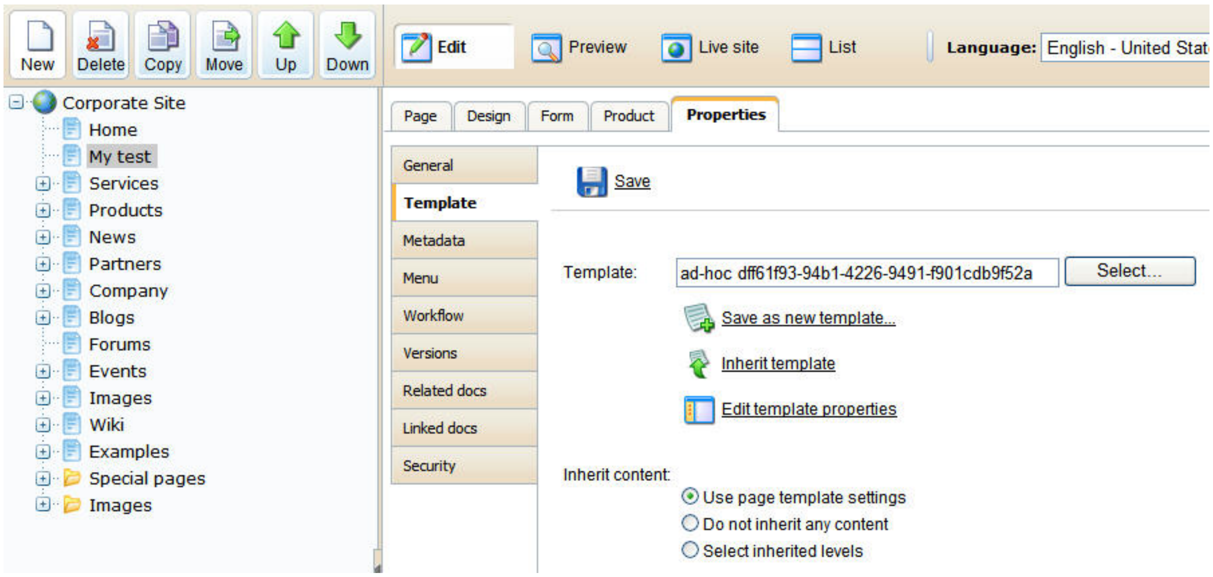

**What you did**

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 that enables content editors to enter text into the page.

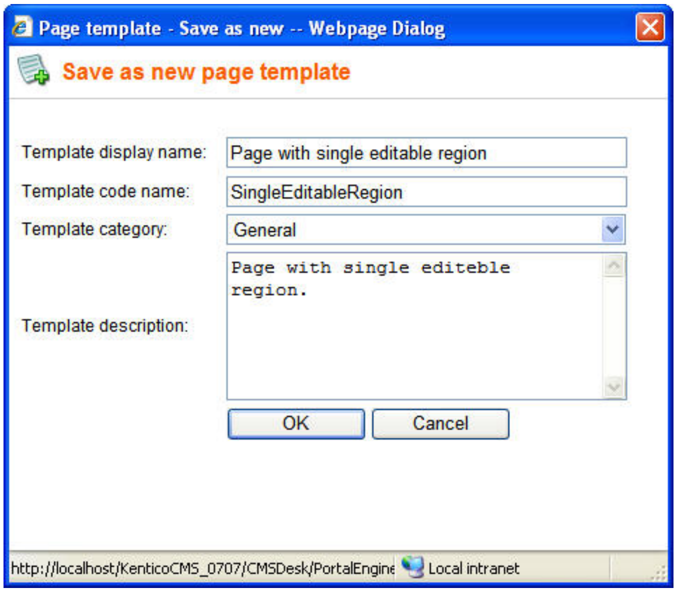
## 10.3 Re-using an ad-hoc page template

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**.

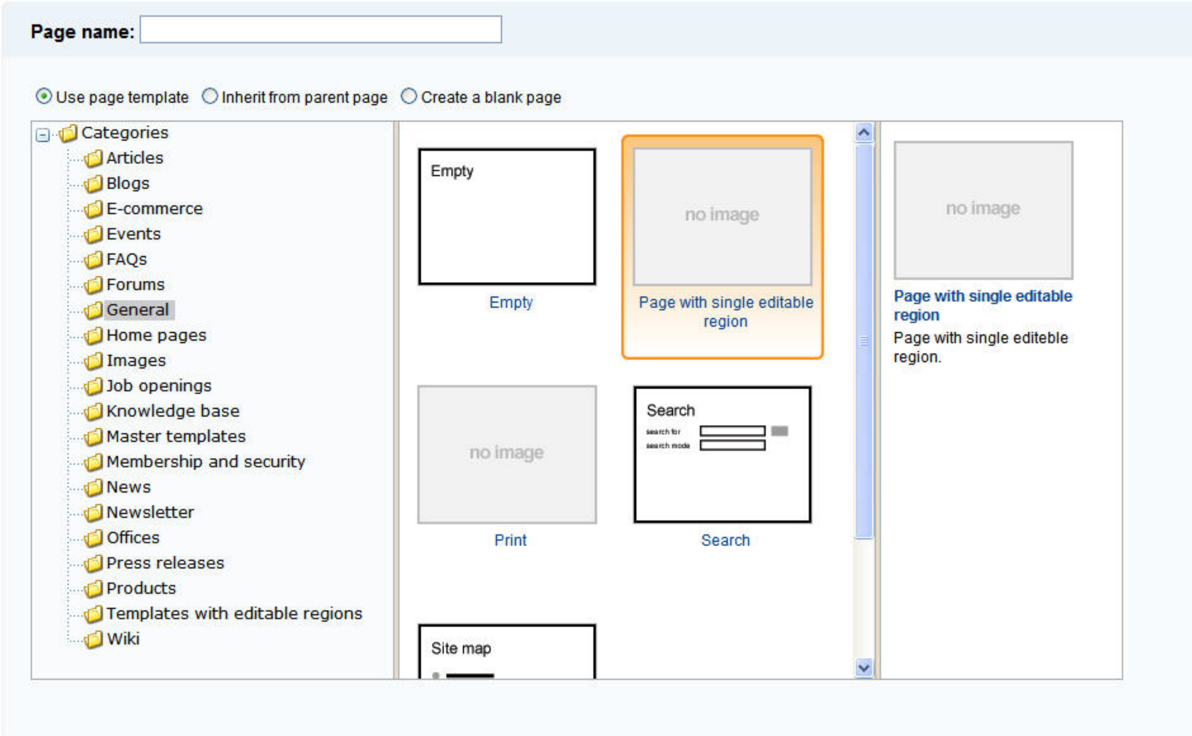
Go to **Content** -> select your new page created in the previous step -> click **Properties** and click **Template**. You will see a page like this:



As you can see the current page is based on an ad-hoc template with some unique identifier. 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 on the picture below and click **OK**, then click **Save** on the **Template** dialog.



Now when you try to create a new page, you will be able to create the page based on this new page template:



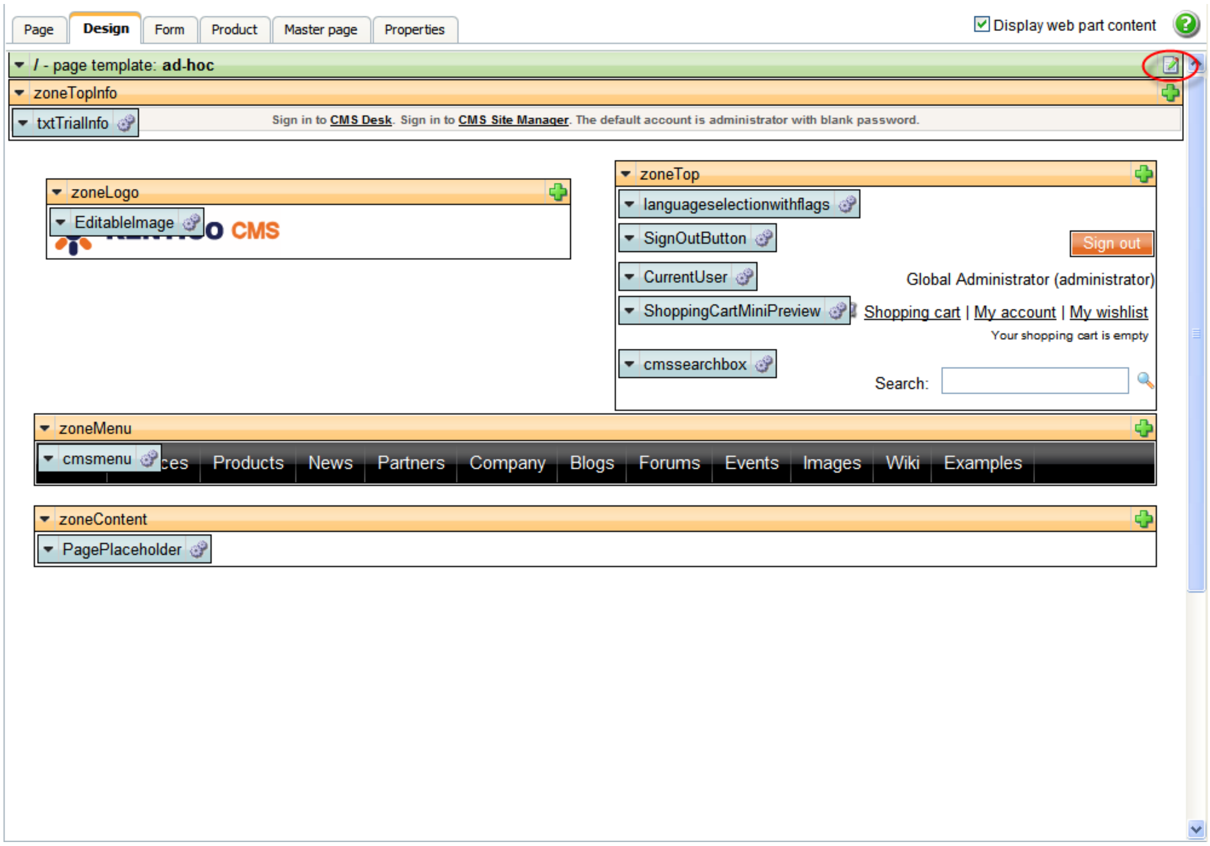

### What you did

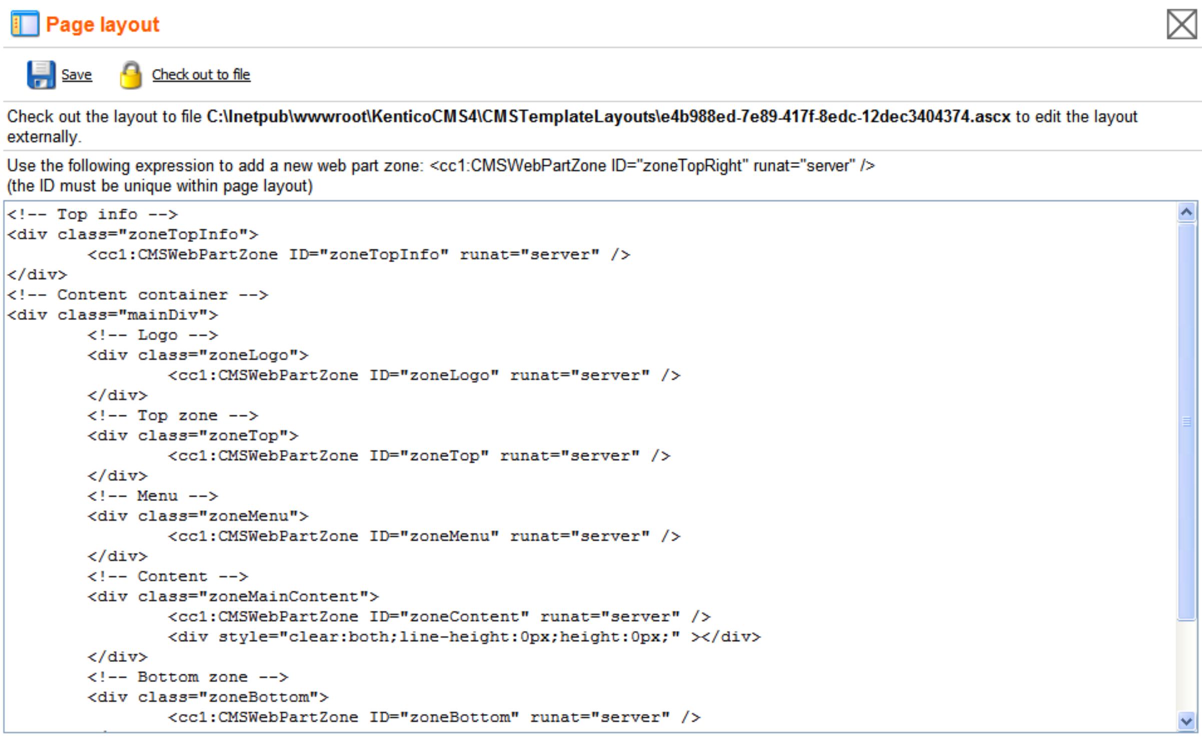
You have converted the previously created ad-hoc page template that was specific only for a single page into a re-usable page template that can be used for many pages.

## 10.4 Page layouts

A **page layout** consists of **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 how to edit a page layout is to switch to the **design mode** 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 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.

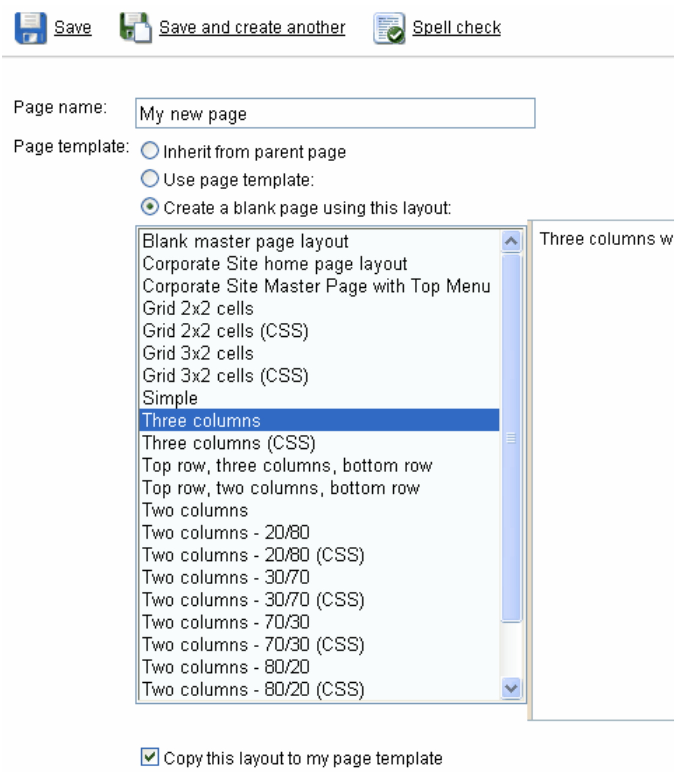
**Editing the layout in your favorite editor**

Editing of HTML code in plain text area may not be very comfortable. However, you can use the button **Check out to file** 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 site displayed to the users. |
| Layout code name | The name of the site used in the code. |
| Layout description | Optional description. |
| 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>
```

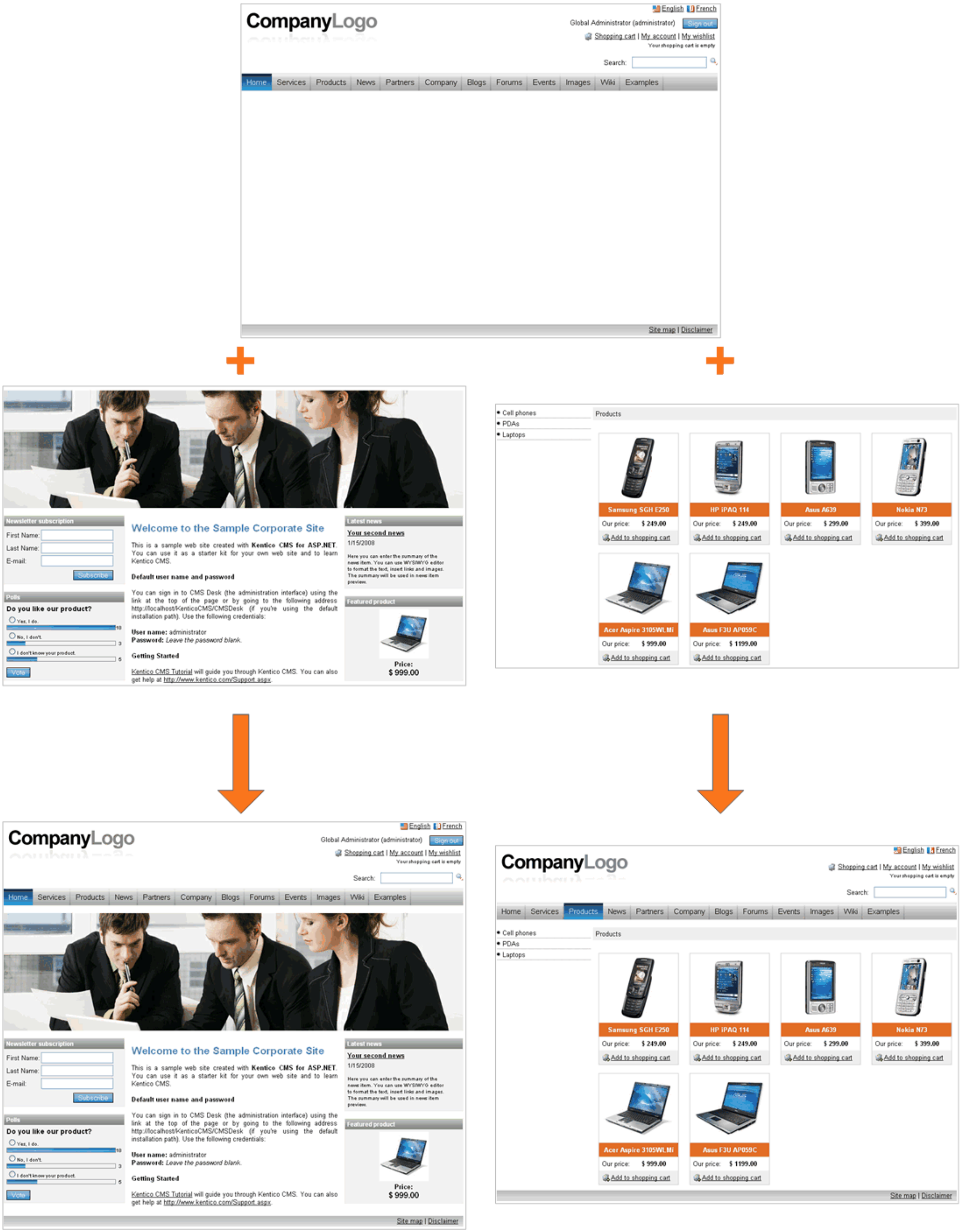
The following layout code defines the same two-column layout, but using DIV elements and CSS styles:

```
<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>
```

## 10.5 The master page concept

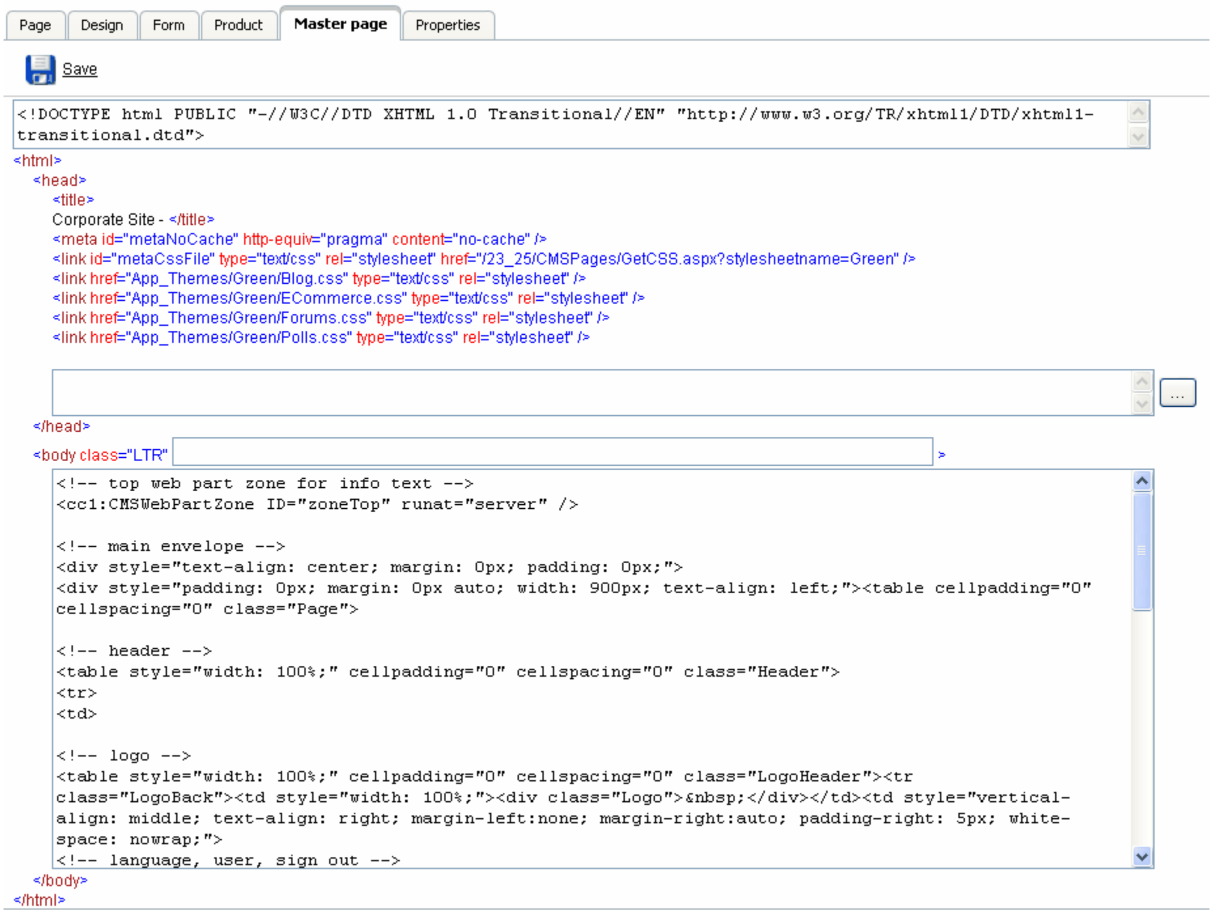
Master pages represent a powerful concept of sharing the same header and footer for all pages on the web site. It allows you to manage repeated items, such as site logo, main menu and footer content on a single place.

The following figure shows how the same master page is used for home page and for product page. As you can see, the pages are inserted inside the master page:



## 10.6 Editing the master page

The master page is represented by the root document of the tree structure. It can be edited as any other page. You can use the **Design** tab to edit the web parts and HTML layout of the master page. Besides, there's a special **Master page** tab available only for the master page:




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 on your web site:

- **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 in the **<head>** section 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 edit on the **Design** tab, in the **HTML Layout** mode.

**Page placeholder**

The 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 chapter.

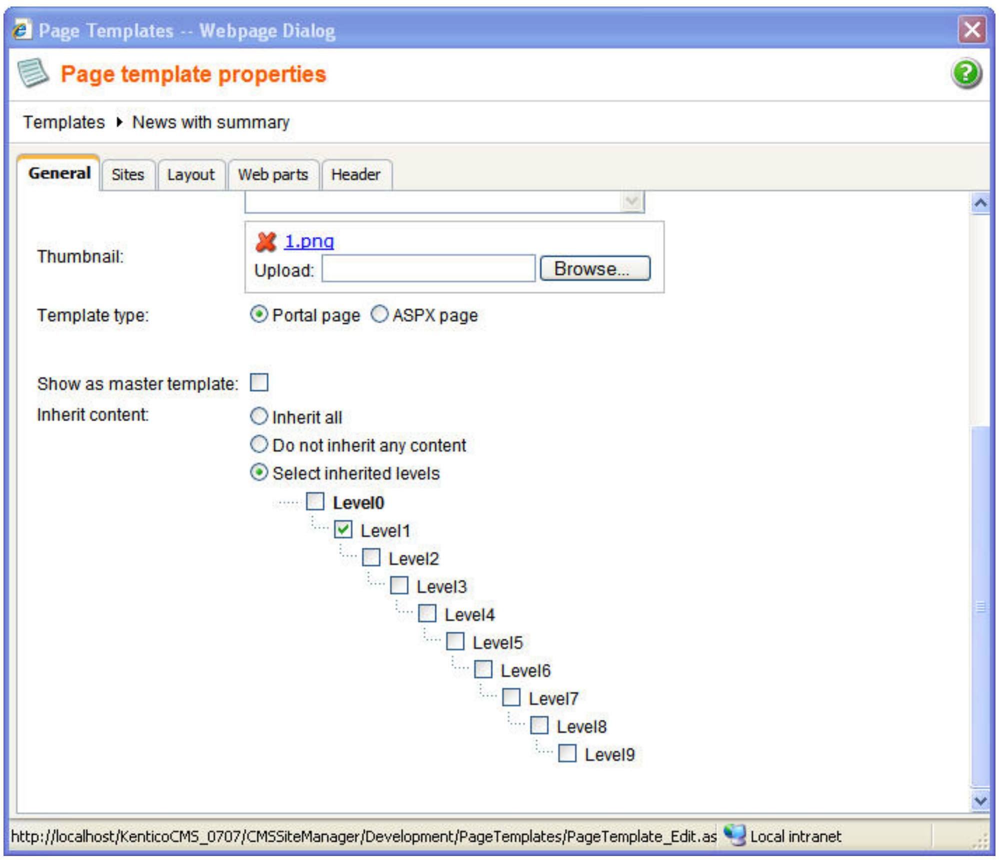
# 10.7 Visual inheritance

As you saw in chapter The master page concept, the content of sub-pages is displayed within master page or generally within any parent page using the **Page placeholder** web part. The impact of this approach is that the sub-page content is "nested" inside the content of the parent pages.

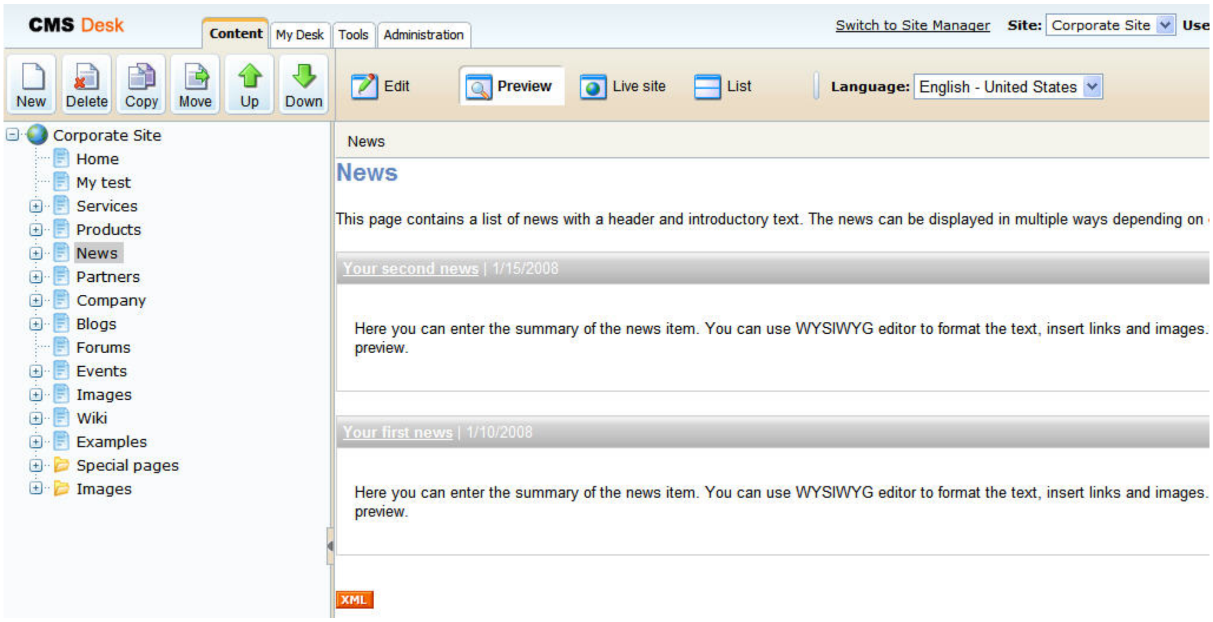
In some cases, you may want to hide some parts of the parent page. There are several ways how to achieve that:

## 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 **Level 1** boxes. It means that only the content from first level of content hierarchy will be displayed and the master page (root) is not inherited. Click **OK** to save the changes.

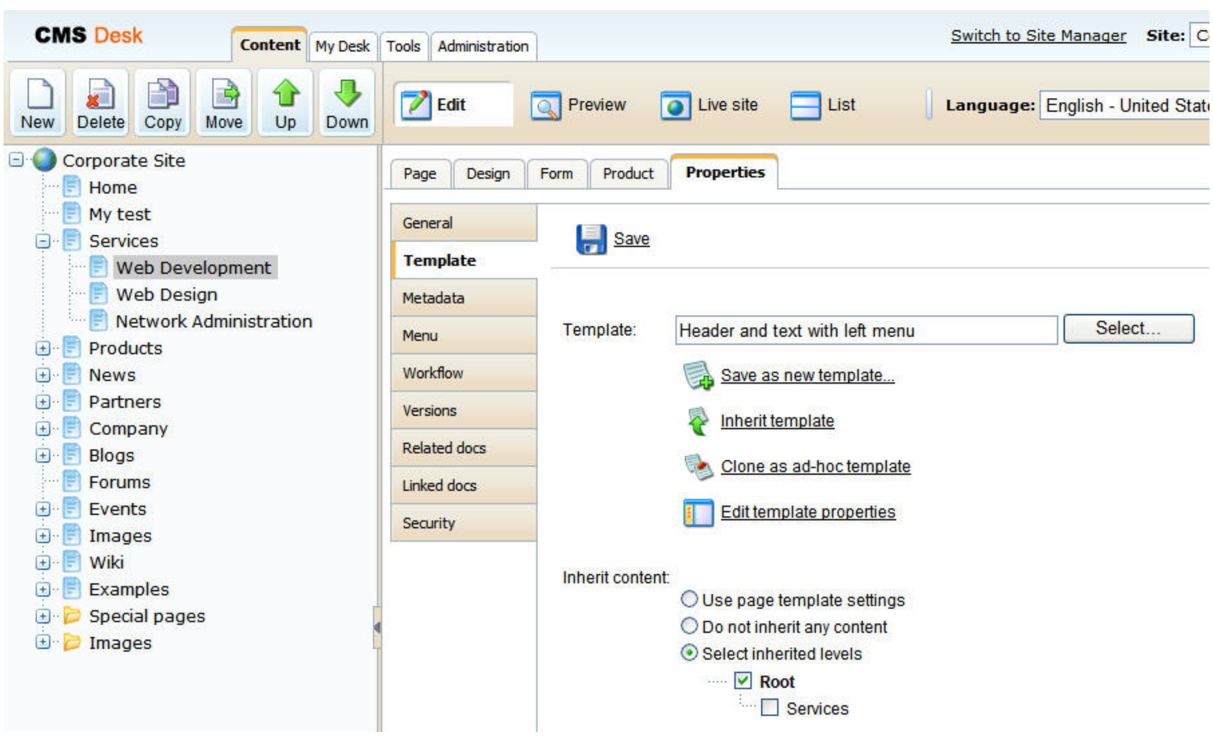


The page will look like this:



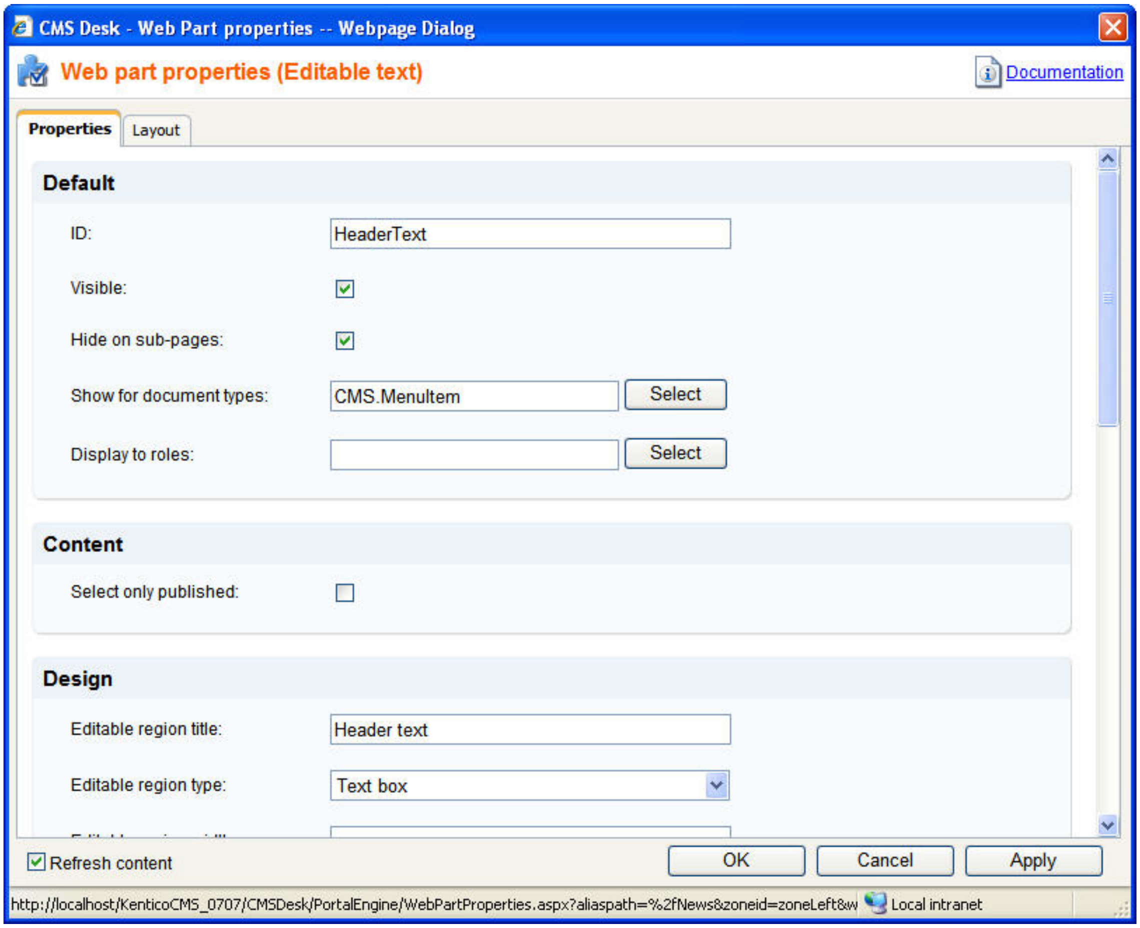
Set the value back to empty value for now.

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

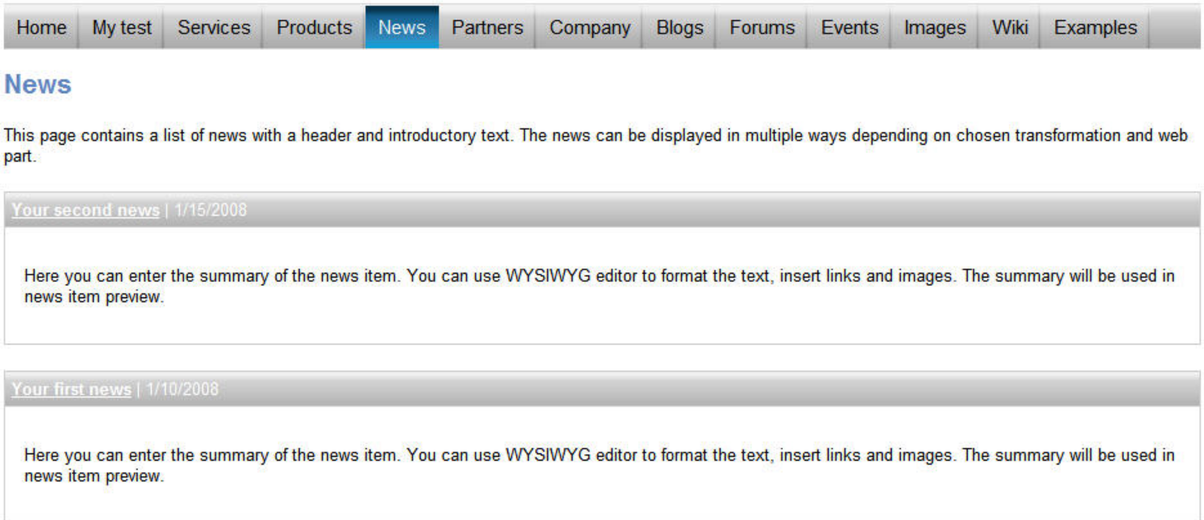


### Using the "Hide on sub-pages" web part property

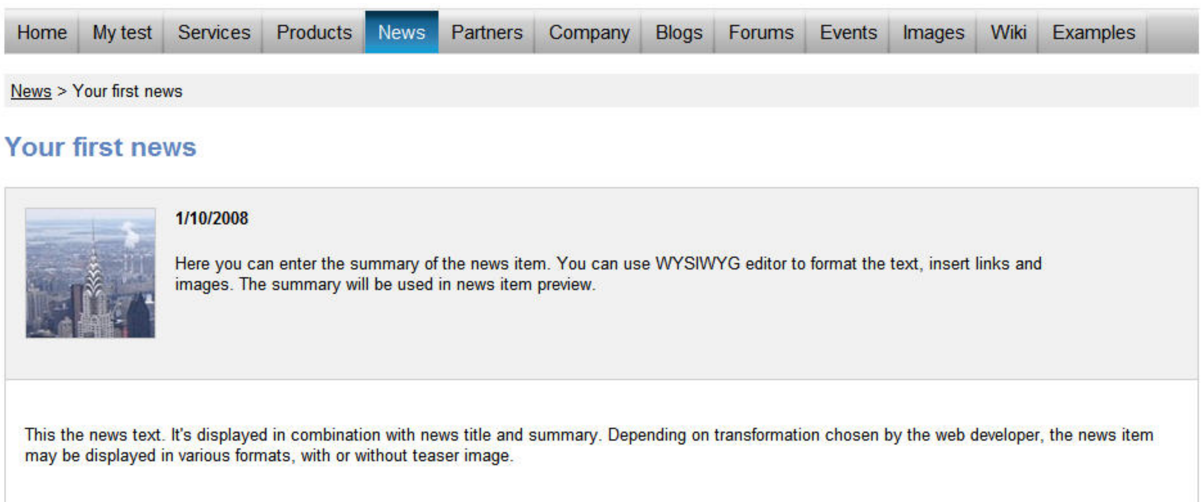
Click **/News** and click **Design**, click **configure** in the **HeaderText** web part. The web part has the property Hide on sub-pages set to true:



Click **Cancel** and click **Live site**. Please note that when you display the list of news, the title **News** is displayed:



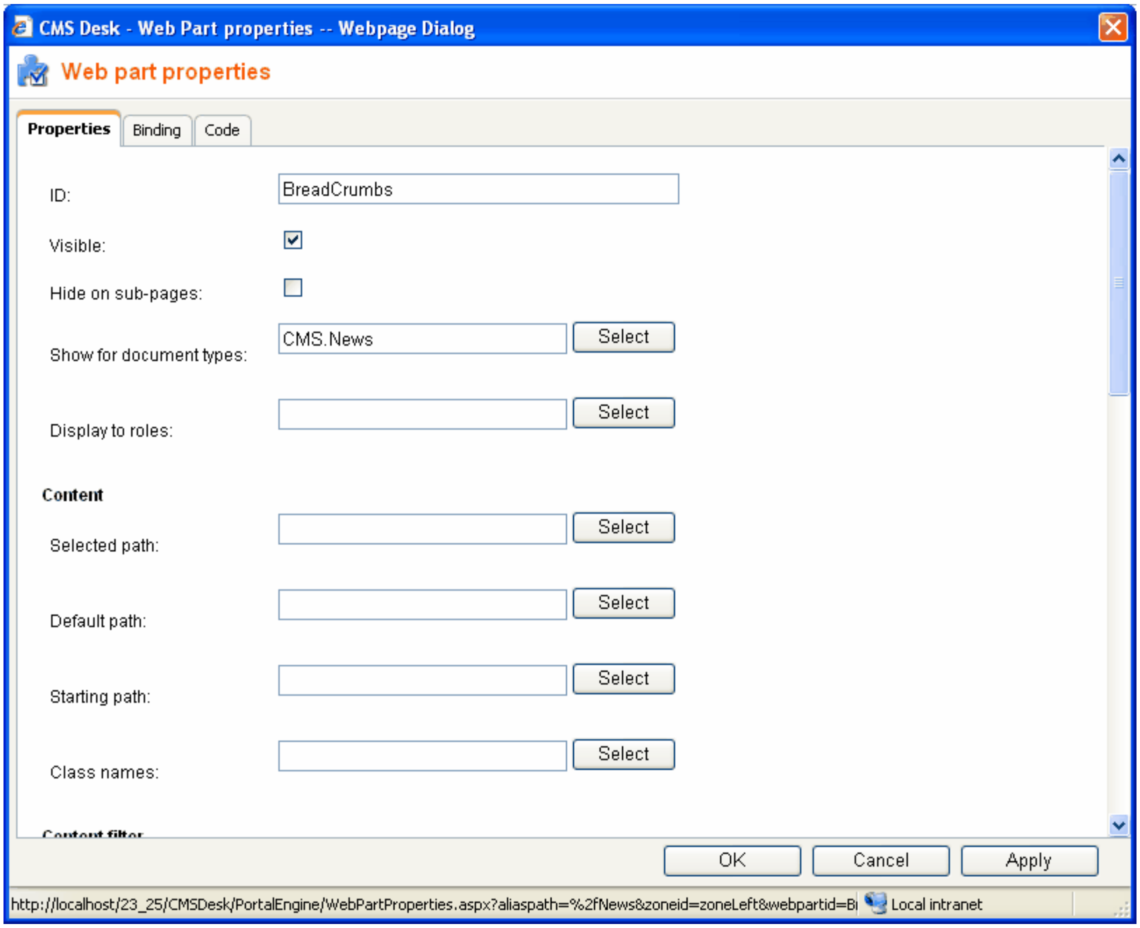

If you go to some particular news item, the title is hidden:




This is ensured by the **Hide on sub-pages** property that hides the web part displaying the **News** title.

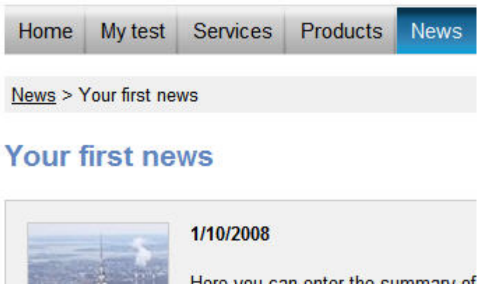
## Using the "Show for document types" web part property

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:

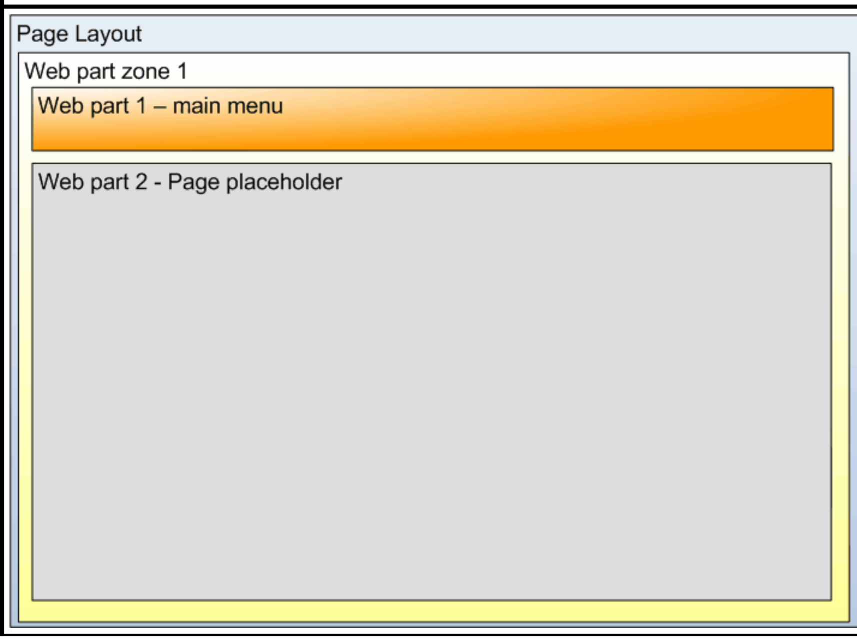
When you go to some particular news item, the currently selected document is a news item and the breadcrumb navigation is displayed:

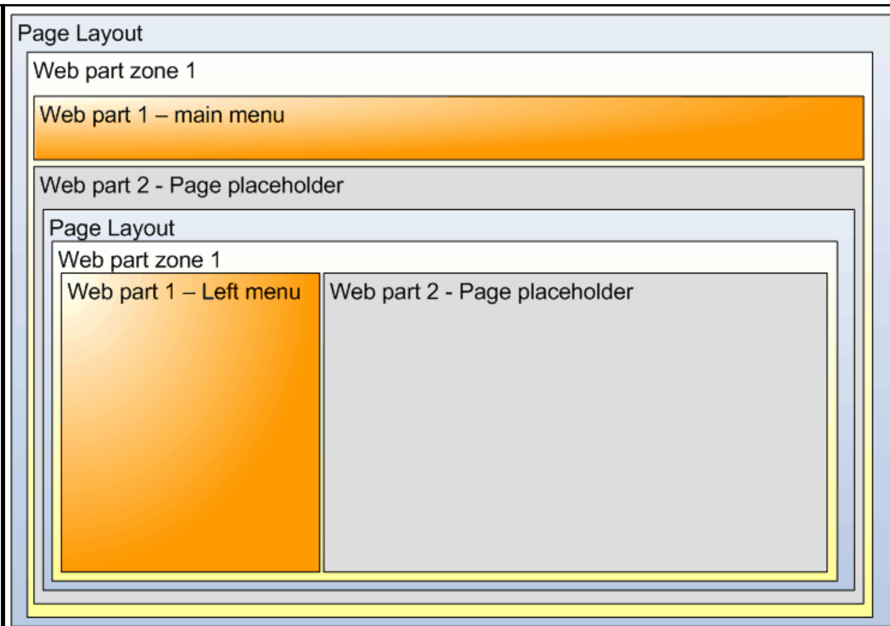
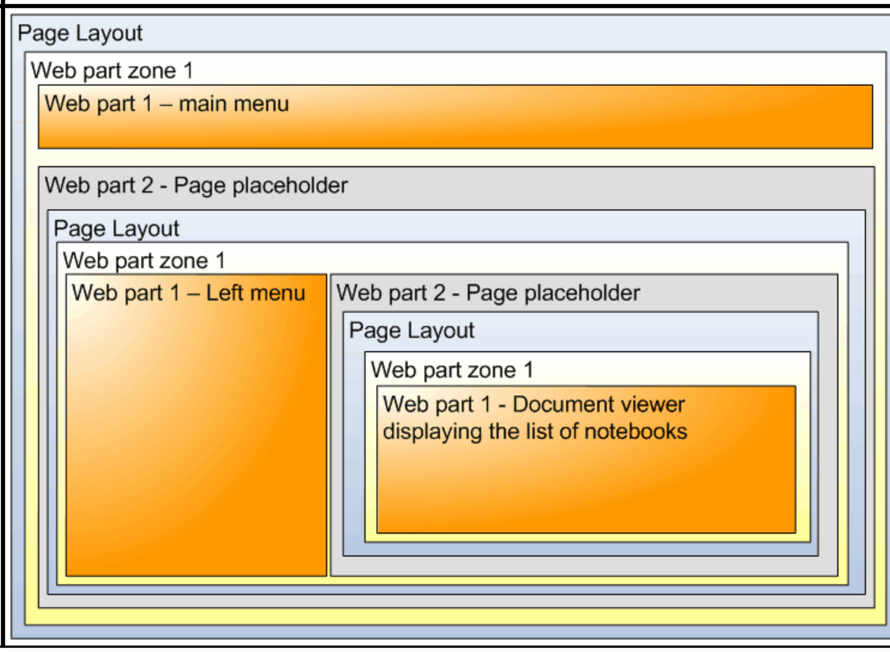


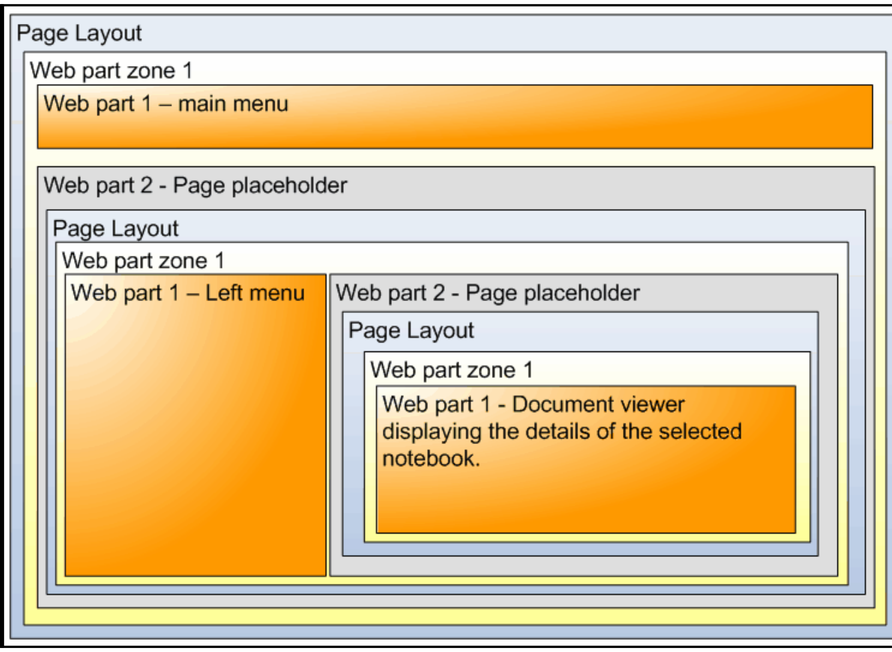

You have learned how to break inheritance of content or how to display content based on the current document type.

## 10.8 Content tree and page templates

The content tree defines not only the site map of your web site, but it also defines how the pages are nested. When you request a page **`/products/notebooks/dell800`**, the portal engine works loads the content in the following order:

| Processed path | Page template | Rendered page |
|---|---|---|
| **/ (root)** | Main menu page template |  |

<table>
<tr><td>/products</td><td>Products page template</td><td>
</td></tr>
<tr><td>/products/ notebooks</td><td>Notebooks page template</td><td>
</td></tr>
</table>

| | | |
|---|---|---|
| **/products/notebooks/dell800** | *Inherits page template from parent page. The product detail is displayed automatically instead of the listing by the document viewer.* |  |

## 10.9 Managing page template catalog

You can manage the page templates in **Site Manager -> Development -> Page templates**. Here you can organize the templates into categories, modify their descriptions, etc.

Each template has the following properties:

### General tab

| | |
|---|---|
| Template display name | The name of the template displayed to the users. |
| Template code name | The name of the template used in the code. |
| Category | Here you can choose the category of the page template. |
| Template description | Description. |
| Template type | Indicates if the page template is a portal page template or ASPX page (the later option is used for backward compatibility with CMS 1.x). |
| Show as master template | Indicates if the field should be offered in the New site wizard among master page templates. Master page templates are usually used for main menu and logo of the web site. |
| Inherit content | Configures the visual inheritance as explained in the Visual inheritance chapter.<br><br>You can choose from the following options:<br><br>• **Inherit all** - inherits the content of all parent pages<br>• **Do not inherit any content** - displays only the current page without any parent content<br>• **Select inherited levels -** displays only content from the chosen content tree levels. |

### Sites tab

On the Sites tab, you can choose which web sites the page template will be available for.

### 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").

### Web parts

Here you can see the XML document with configuration of the web parts. Use this dialog only in situation when the standard **Design** dialog doesn't work as expected (e.g. due to some data inconsistency).

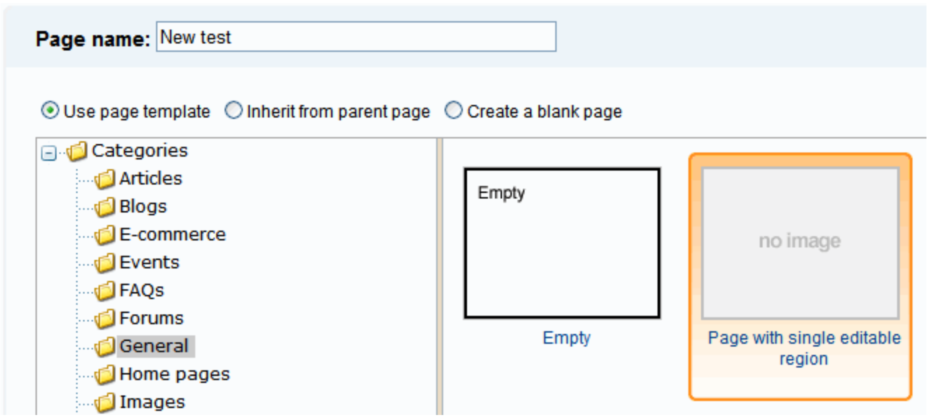
### Header

Here you can specify custom HTML code that will be added to the <head> element of the page for all pages that use this page template. It's useful if you need to link some additional CSS or JavaScript files.

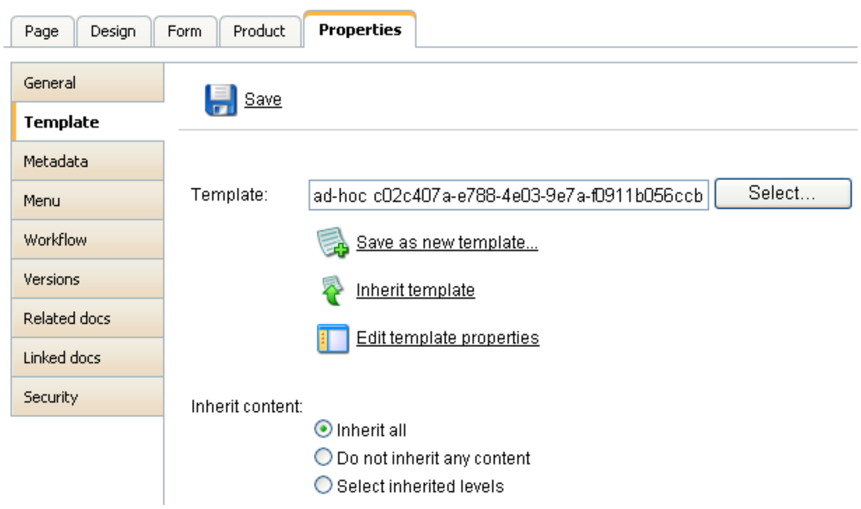
## 10.10 Cloning and modifying a page template

You may need to create a new page that will be similar to existing pages, but with some minor modification of the page template. Consider you want to display the editable region from the previously created page template in a green container box.

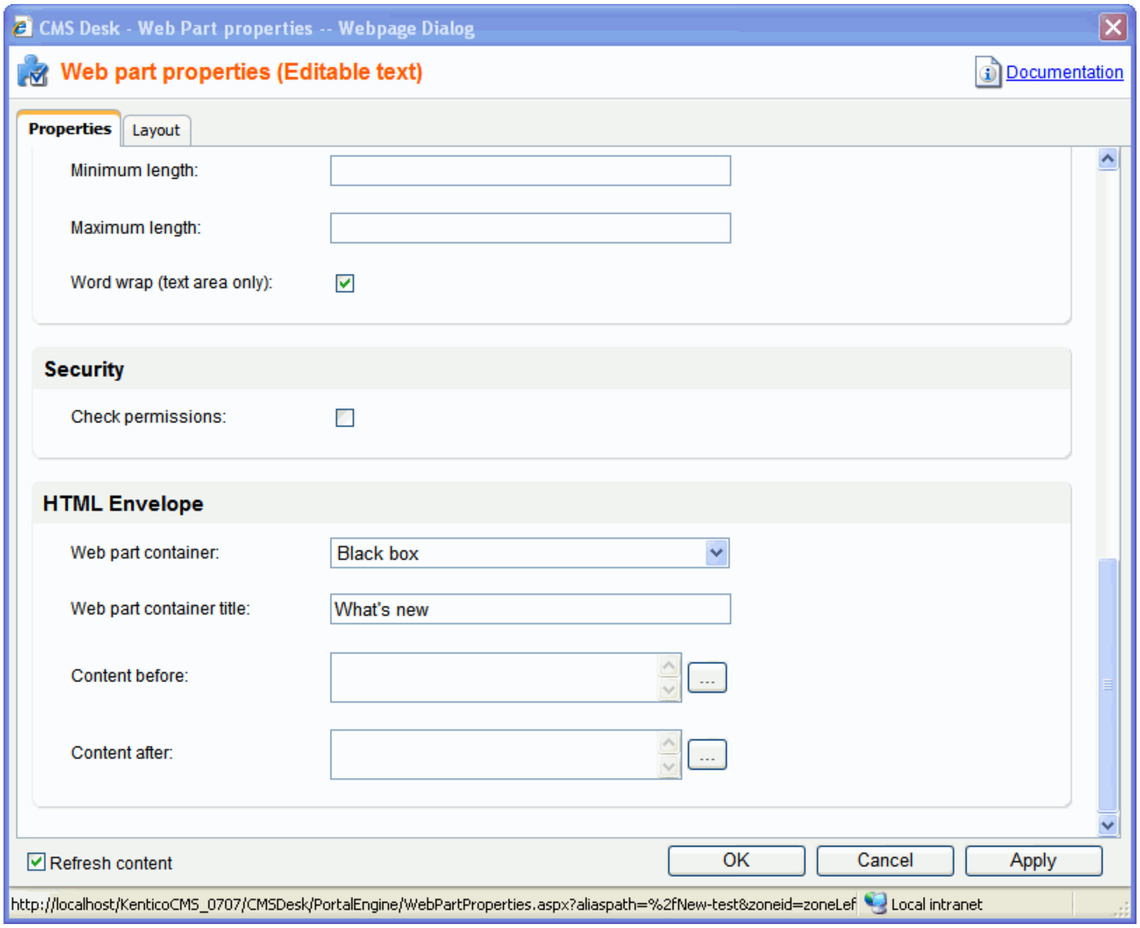
Instead of creating the page template from scratch, simply create a new page based on your existing page template.



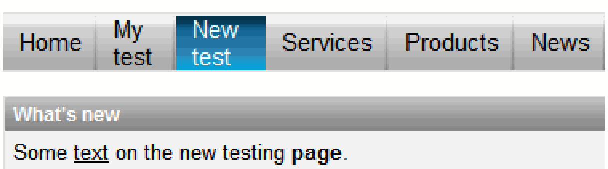

Then, choose the page and click **Properties -> Template** and click **Clone as ad-hoc template** and click **Save**. A new ad-hoc template is created now and you can edit it without modifying the existing pages based on the original template:



Now switch to the **Design** mode and click the **configure** button of the Editable text web part. Choose to use the **Black box** container in the **Use container** value and enter **What's new** into the **Container title** field:




Click **OK**. Switch to the **Page** mode 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 with a green 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**

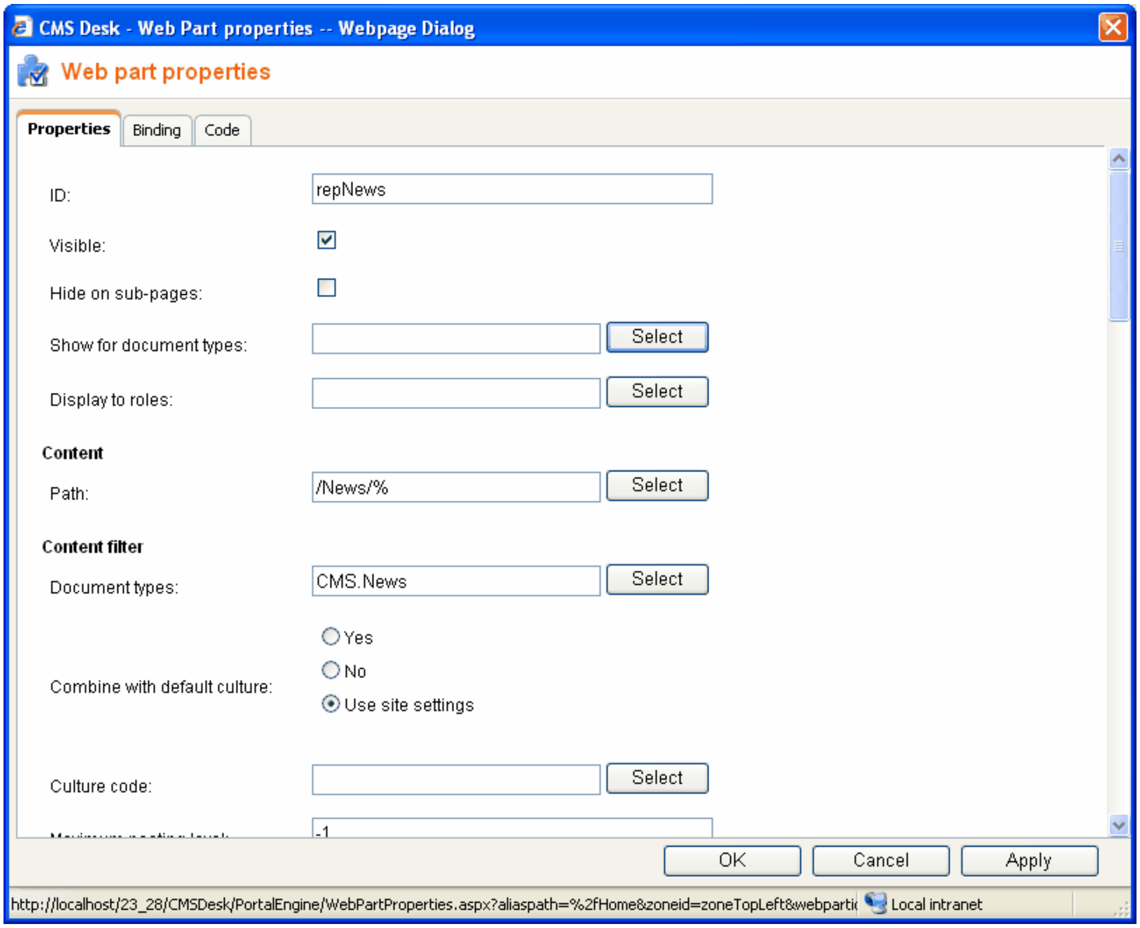
If you want to re-use the new, adjusted page template, you can click the **Save as new template** button in the **Properties -> Template** dialog as you previously did.

## 10.11 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're loosing the friendly user interface and you 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.

## 10.12 Web part binding (obsolete)

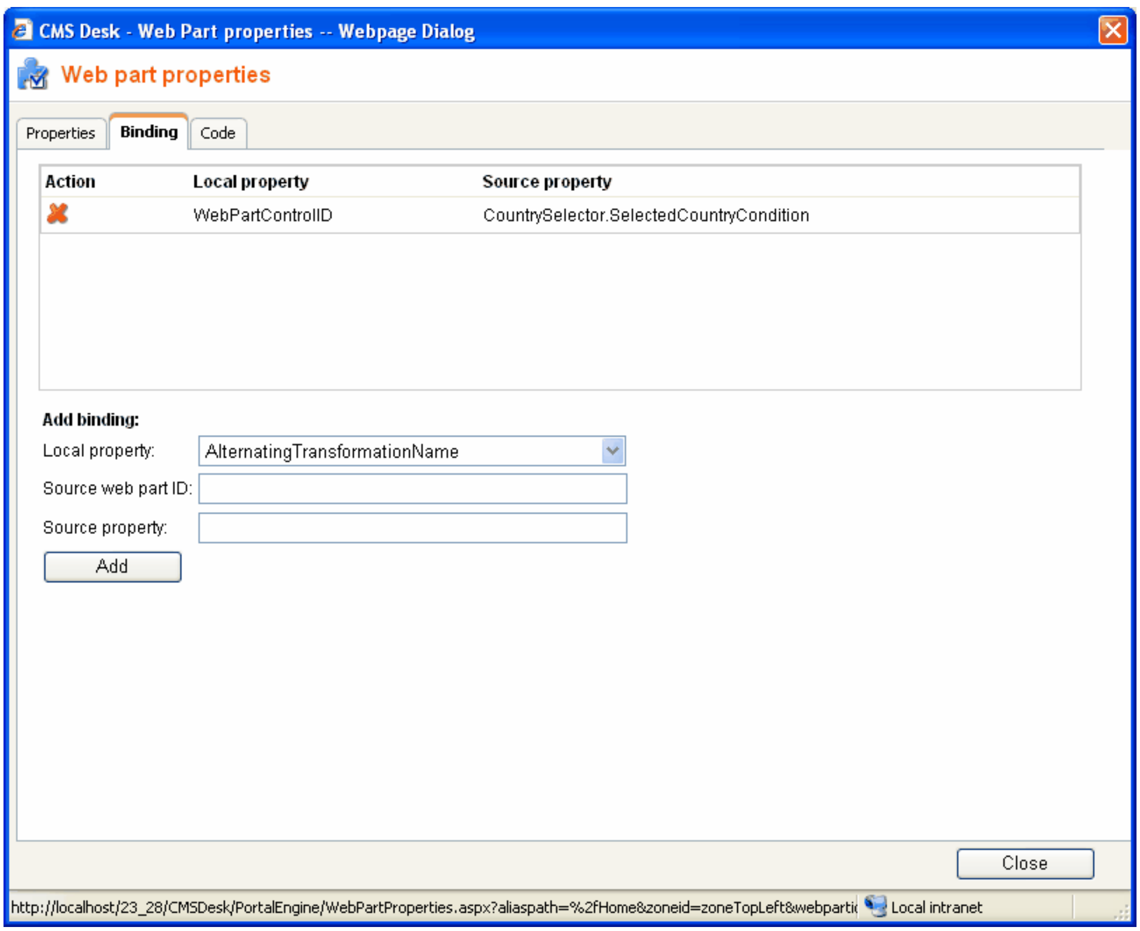
**Obsolete feature**

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 where you put both web parts and write custom code that will ensure the communication between 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 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:



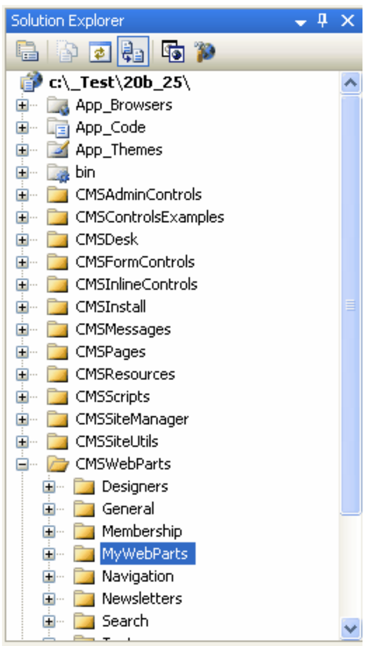
**Alternative option**

If binding doesn't work as expected, you can alternatively create your own user control, place 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 chapter Adding custom code to the page.

## 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 VS2005 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:

```
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;
```

Change the class inheritance like this:

```
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** mode 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.

## 10.13 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 chapter Modifying web part behavior.

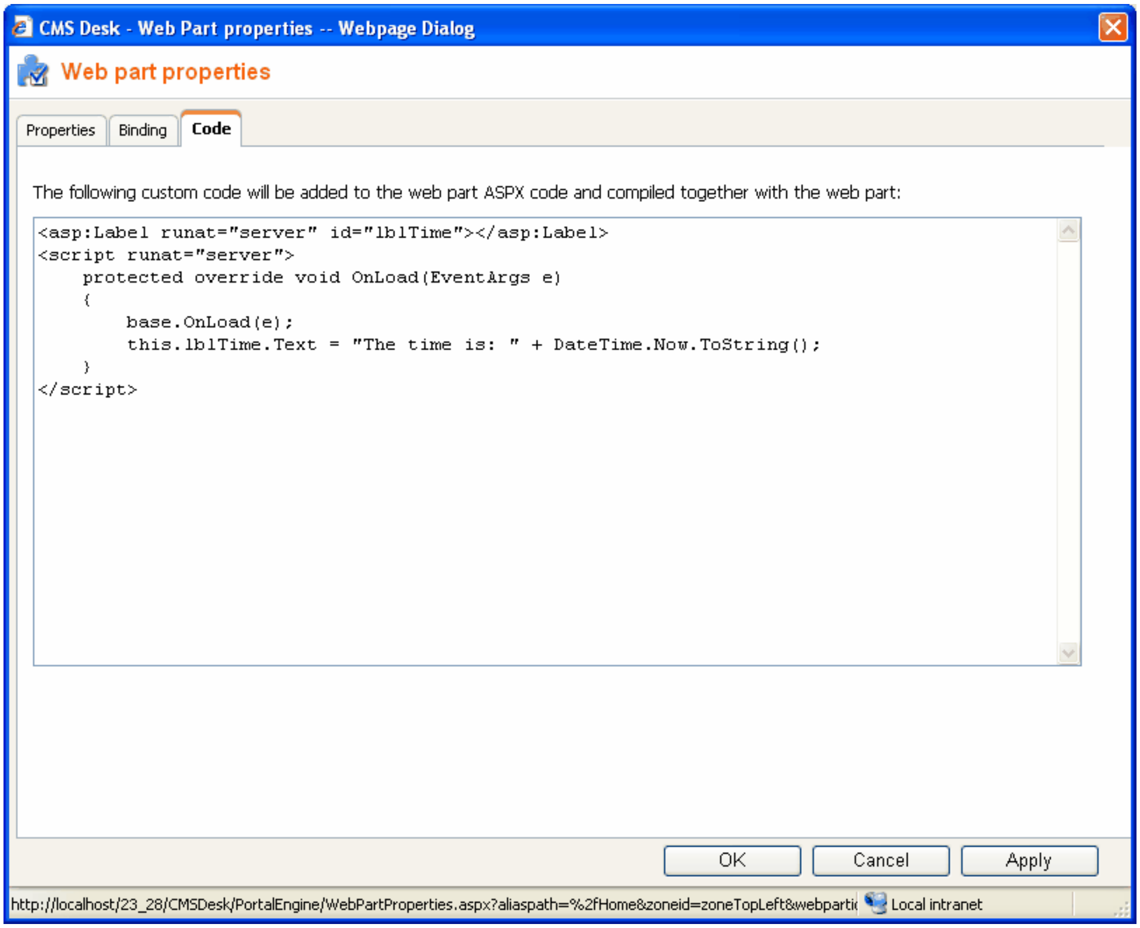
If you still need to use this feature (for backward compatibility), you need to enable it by adding the following parameter to your web.config file, to the appSettings section:

<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 web site and you will not be able to use the web site in medium trust environment. If this is an issue for you, you can use an alternative solution described in chapter Modifying web part behavior.

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 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 use only 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 call also the original method (using base.Method in C# or MyBase.Method in VB). If you omit this, the web part may not work properly.

## 10.14 Common web part properties

Many of web parts use the same or similar properties. The following table summarizes the most important properties found in most web parts:

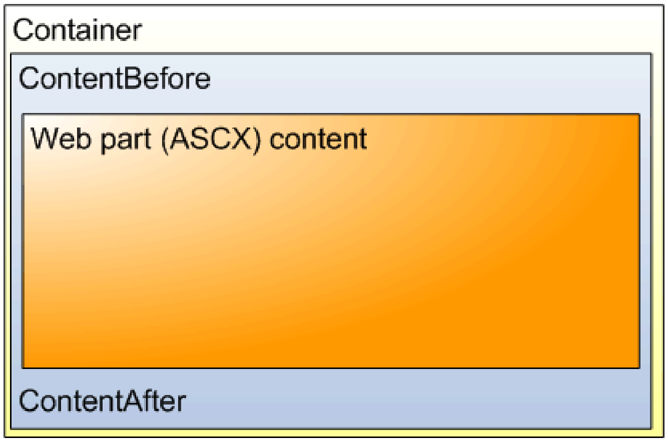
### General properties

| Property Name | Description | Sample Value |
|---|---|---|
| Hide on sub-pages | Indicates if 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. | |
| ID | Web part unique identifier. | text1 |
| Visible | Indicates if web part is visible. | |
| Show for document types | Indicates for which document types should the web part be displayed. If the type of the currently selected document is different, the web part is not displayed. | cms.news;cms.event |
| Display to roles | The list of roles who should see the web part. If now role is specified, the web part is displayed to all roles. | cmseditors;customrole |

**HTML envelope**

| Property Name | Description | Sample Value |
|---|---|---|
| Web part container | Container displayed around the web part. The containers can be defined in **Site Manager -> Development -> Web part containers**. | |
| Web part container title | Title displayed in the web part container (if it supports displaying of title). | Latest news |
| Content before | Text to be displayed before the web part. | <table style="background-color: red"><tr><td> |
| Content after | Text to be displayed after the web part. | </td></tr></table> |

The structure of the web part and its envelope looks like this:




You can find more details on web part containers in chapter Containers overview.

**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 the 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 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 with a semicolon (;).<br><br>In case of menu and navigation web parts, the page (cms.menuitem) documents are selected by default.<br><br>The other web parts retrieve all documents without their custom fields by default.<br><br>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). | cms.article;cms.news;cms.menuitem |
| Combine with default culture | Indicates if default language version of the document should be displayed if the document is not translated to the current language. | You can choose from yes and no or you can choose to use web site-level settings. |
| Culture code | Culture version of the displayed content. | en-us |
| Maximum nesting level | Maximum nesting level. It specifies the number of sub-levels in the content tree that should included in the displayed content.<br><br>Value 1 indicates that only the current document should be returned.<br>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 web site from which you want to display the content.<br><br>If you leave the value empty, the content is retrieved from the current web site. | 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 permissions should be checked for the displayed content and only the documents for which the current user has the "read" permission should be displayed. | |
| Cache item name | Name of the cache item. If not explicitly specified, the name is automatically created based on the control unique ID.<br><br>You can assign the same value of this property if you want to display the same content on several pages and do not want the system to keep the same items in memory redundantly. | latestnewslist |
| Cache minutes | The number of minutes for which the content is kept in the cache until the latest version is reloaded from the database.<br><br>If you specify 0, the content is not cached.<br><br>If you specify -1, the site-level settings are used. | 10 |

**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 | Sub-menu indicator image | /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. | |
| Page size | Number of records per page. | 10 |
| Pager position | Position of the pager - top or bottom | |
| Show first and last buttons | Indicates if First and Last buttons should be displayed. | |
| Query string key | The name of the URL parameter that will contain the current page number. | mylistpage |
| Paging mode | Type of paging parameter - it can be passed either through URL (Query string) or through postback (Postback). | |

**Relationships**

These settings allow you to configure the web part so that it displays only content that is in relation with given (main) document.

| Property Name | Description | Sample Value |
|---|---|---|
| Main document | Document for which you want to display its related documents. | |
| Relationship name | Name of the relationship between documents. | Is related to |
| Main document is on the left side | Indicates if the given document is on the left side of the relationship. | |

**No data behavior**

| Property Name | Description | Sample Value |
|---|---|---|
| Hide if no record found | Indicates if content should be hidden when no record is found. | |
| No record found text | Text that should be displayed if no data is found. | No data found. |

**Editing buttons**

| Property Name | Description | Sample Value |
|---|---|---|
| Show New button | Indicates if the button for adding new items should be displayed in the editing 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 item in the editing mode. | |

**Transformations**

| Property Name | Description | Sample Value |
|---|---|---|
| Alternating transformation | Transformation used in the list view mode for even items. | cms.news. preview_alternating |
| Transformation | Transformation used in the list view mode. | cms.news.preview |
| Selected item transformation | Transformation used in the detail view mode. | cms.news.default |
| Item separator | Item separator displayed between records. | <hr/> |

## 10.15 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 dynamic value instead of constant in the property. Such as dynamic value is evaluated in run-time and allows you to specify context-dependent values. See Appendix A - Macro expressions for details.

## 10.16 Adding custom code to the portal page template

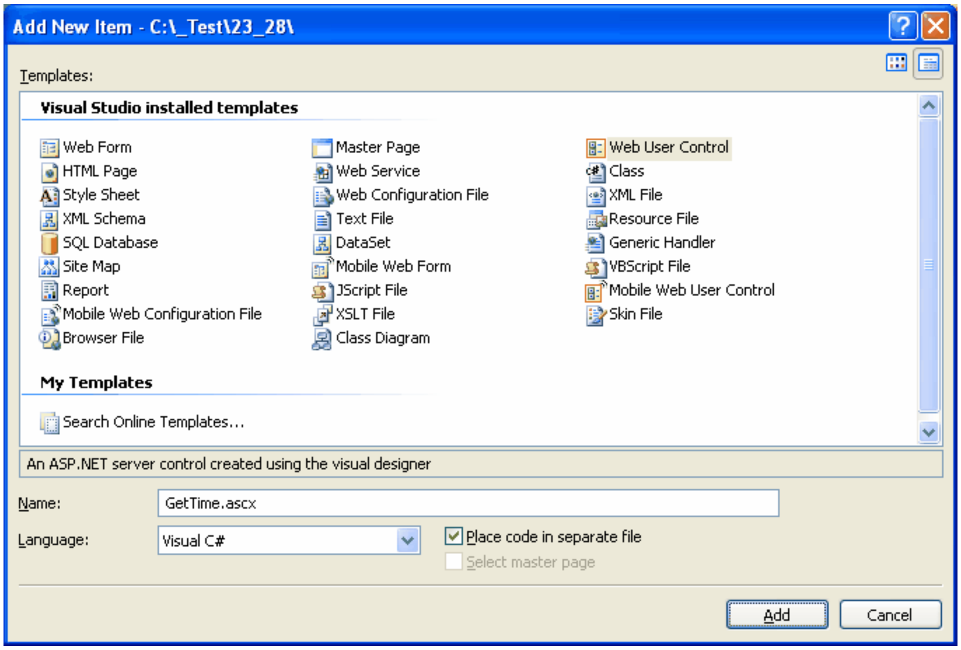
The easiest way how to insert custom code into the portal engine-based web site is using the standard ASCX user controls. This chapter will show you how to do that. If you're not familiar with Visual Studio 2005 development, you can skip this chapter.

### Current time example

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

Open the web site project using the WebProject.sln file that is placed in the folder where you deployed the web site. Right-click the web project in the Solution Explorer window and click **New Folder**. Call the folder 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 the live server.

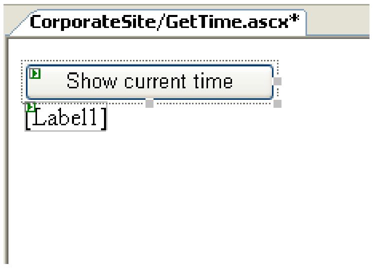
Right-click the new folder and click the **Add new item...** option. Choose to create a new Web User Control and set its name to **GetTime.ascx**. You can set the programming language option to either Visual C# or Visual Basic.




Click **Add**. Switch to the **Design** mode 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 to the Button1_Click method:

**[C#]**

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

**[VB.NET]**

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

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

**Save** all changes.

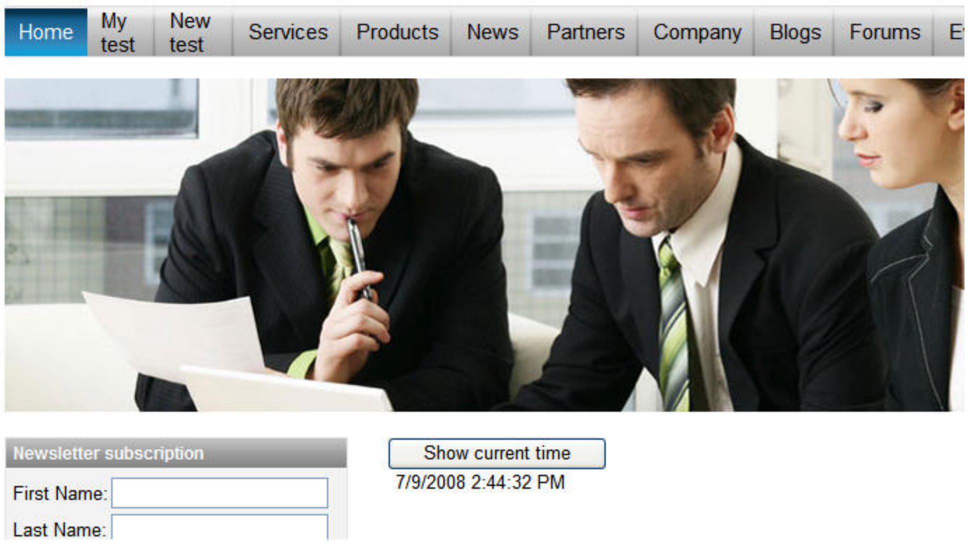
## Adding the user control on the page

Sign in to **Kentico CMS Desk**, click the **Home** page and click **Design**. Remove the web part in the **zoneCenter** zone and click **+ (Add web part)** 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 below 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 2005. 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 build web parts in case that you need to create re-usable, parameterized user controls. The web part development is described in chapter Developing web parts.

## 10.17 Displaying data from external database or Web Service

Beside displaying Kentico CMS content, you can also display data from your external databases or Web Service. In this case, you need to develop a user control (ASPX) 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 if you created the web site from scratch.

### Example: Retrieving data from the sample Northwind database

In this simple example, you will see how to display data from the Categories table of the Northwind database using ADO.NET, in the sample Corporate Site. 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 2005 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 web site later, when you decide to import the web site 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:

**[C#]**

```
using System.Data.SqlClient;
```

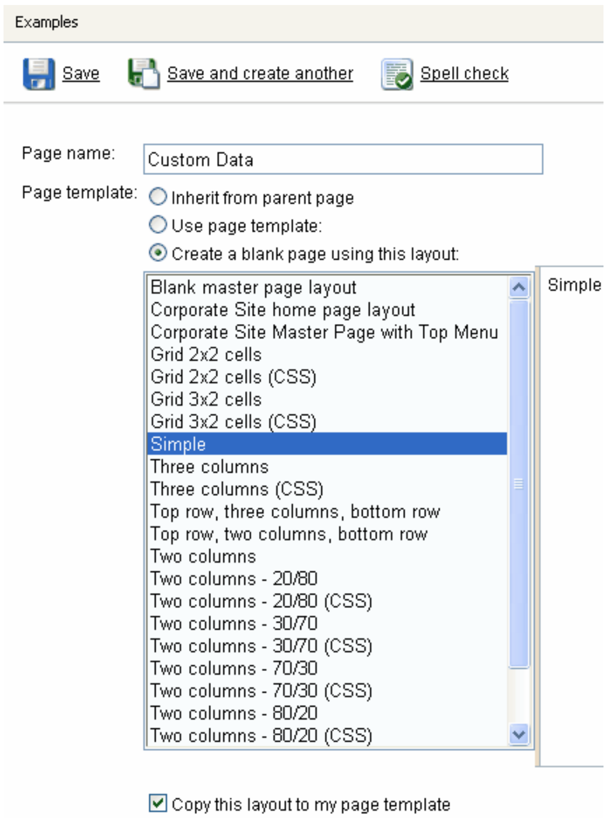
Add the following code to the **Page_Load** method:

**[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();
```

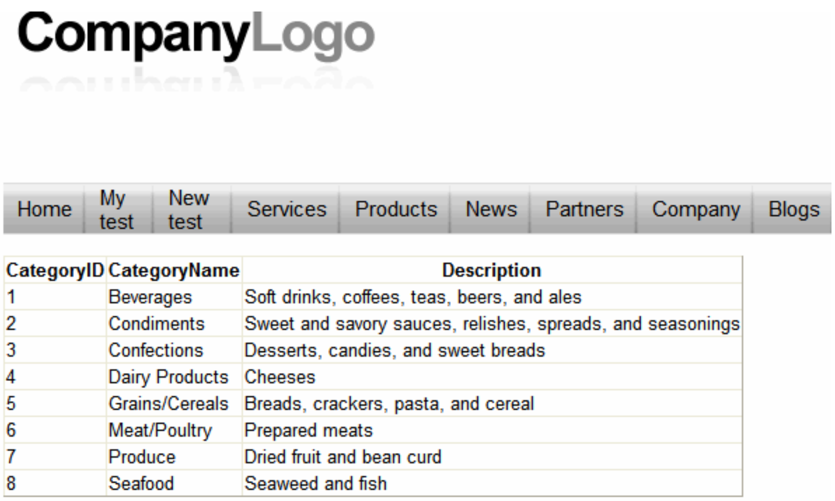
**Save** all changes.

Sign in as administrator to **CMS Desk**, select **/Examples** document in the content tree and create a new **Page (menu item)**. Call it **Custom Data** and choose the **Create a blank page using this layout** option. Select the **Simple** layout and click **Save**.




Switch to the **Design** mode and 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 external database:




| CategoryID | CategoryName | Description |
|---|---|---|
| 1 | Beverages | Soft drinks, coffees, teas, beers, and ales |
| 2 | Condiments | Sweet and savory sauces, relishes, spreads, and seasonings |
| 3 | Confections | Desserts, candies, and sweet breads |
| 4 | Dairy Products | Cheeses |
| 5 | Grains/Cereals | Breads, crackers, pasta, and cereal |
| 6 | Meat/Poultry | Prepared meats |
| 7 | Produce | Dried fruit and bean curd |
| 8 | Seafood | Seaweed and fish |

As you can see we used standard ASP.NET methods to display external data on the web site.

# Part XI

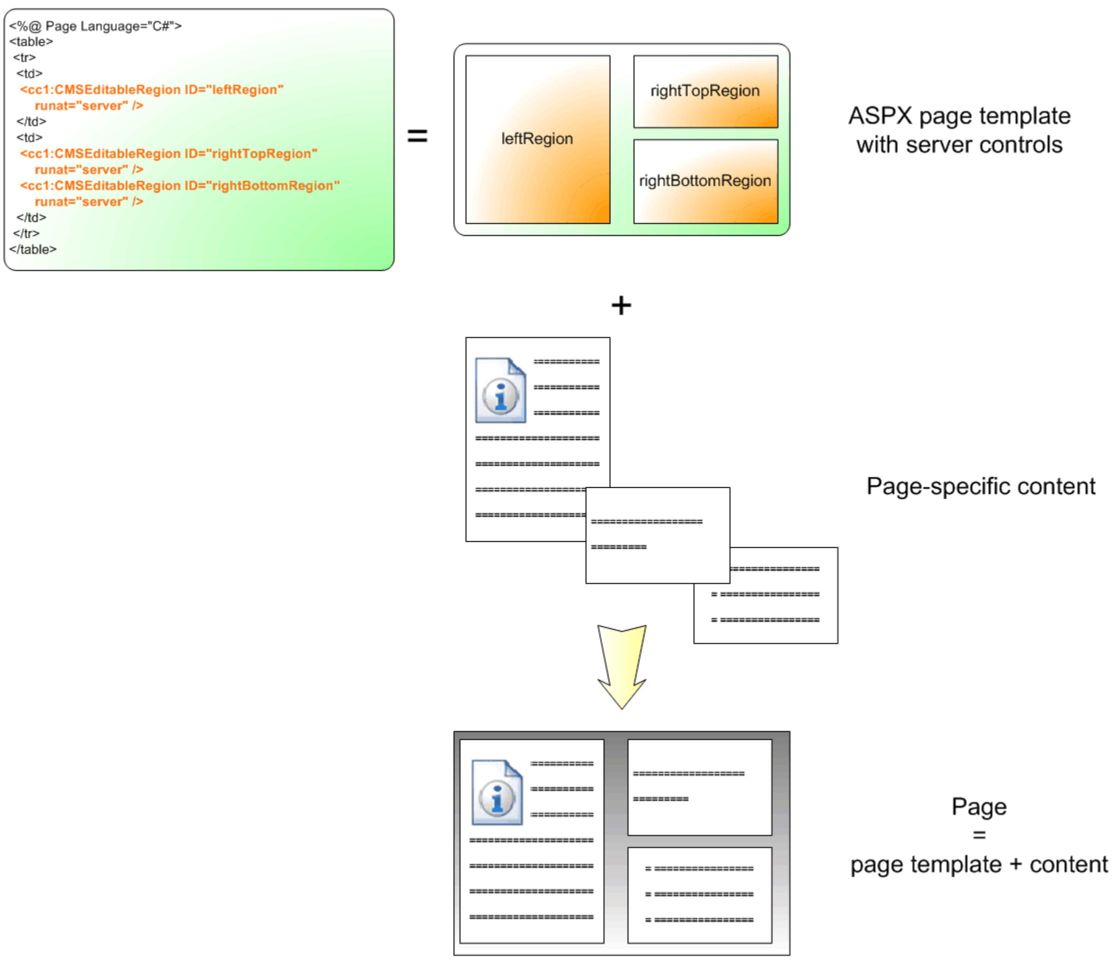
# 11 Development with ASPX page templates

## 11.1 How it works

If you're familiar with ASP.NET development in Visual Studio 2005/2008, you may want to choose to develop web sites using standard ASPX page templates. ASPX page templates in Kentico CMS are standard ASP.NET pages that display content from Kentico CMS. They receive a URL parameter **aliasPath** that tells the page template which page should be displayed.

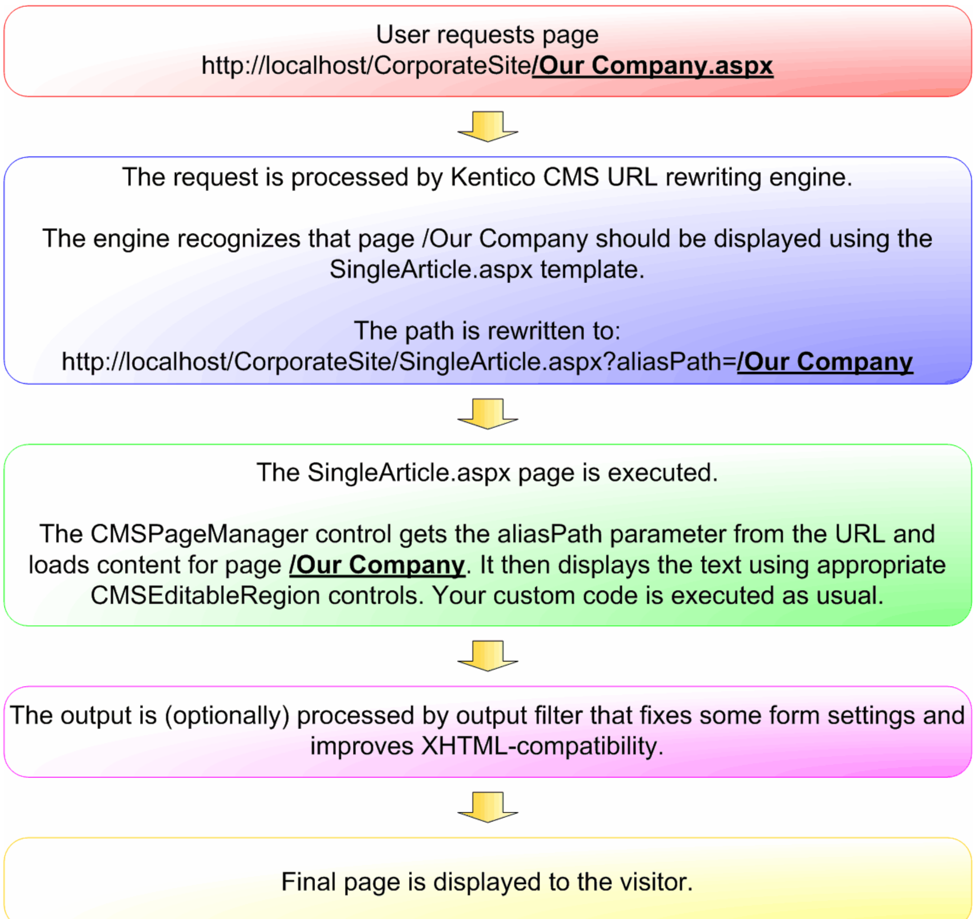
### What does the ASPX page template consist of?

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 (in both VB.NET and C#) to modify page behavior and add custom functionality. The following figure illustrates how 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 URL parameter aliasPath that specifies what content (which page from the content tree) should be displayed using the given template:




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

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

## 11.2 Creating a new ASPX page template

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

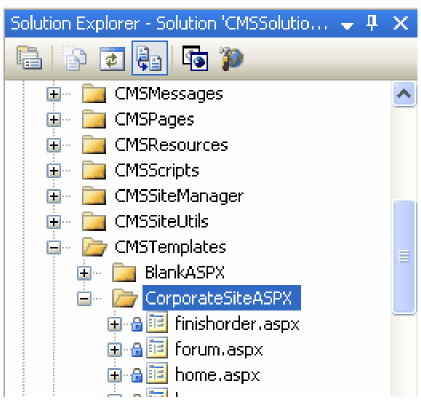
**Adding Kentico CMS Controls to your Visual Studio Toolbox**

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 on the ASPX pages.

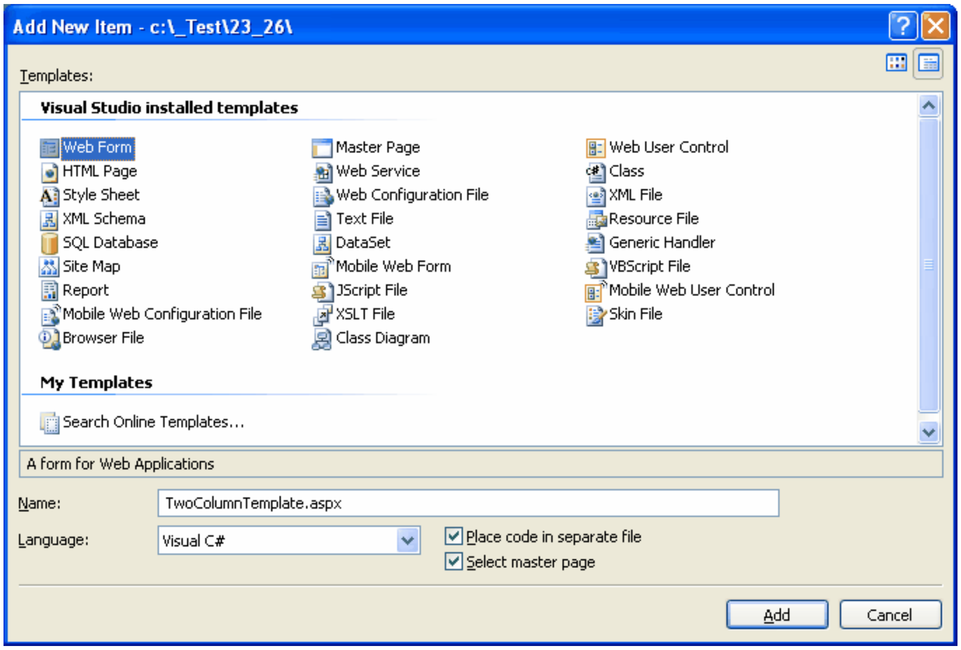
You can find step-by-step instructions in chapter Adding Kentico CMS Controls to the Toolbox.

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

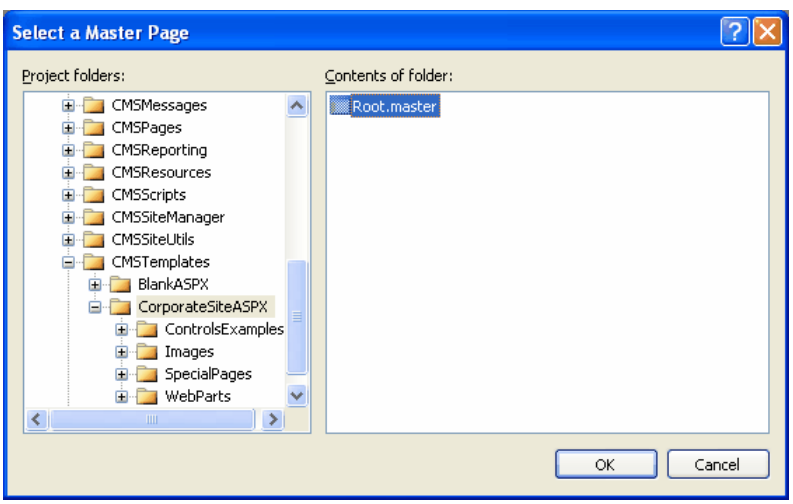
Now right-click the **CMSTemplates -> CorporateSiteASPX** folder in the Solution Explorer and choose to add a new web item:



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




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

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="cc1" %>
```

Then add the following code inside the <asp:Content></asp:Content> control:

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

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

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

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

```
using CMS.UIControls;
```

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

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

to this:

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

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

## Registering the ASPX page as a page template

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

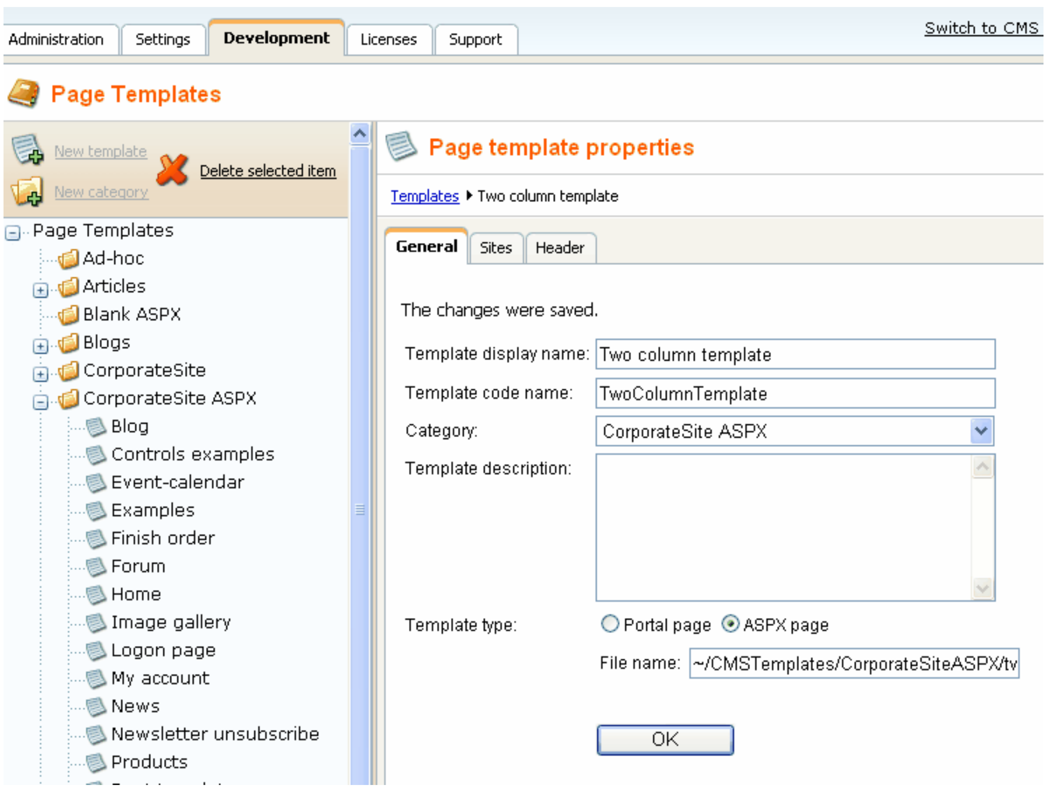
Sign in to **Site Manager** and go to **Development -> Page templates**. Click the **Corporate Site ASPX** folder and click **New template**. Enter the following values:

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

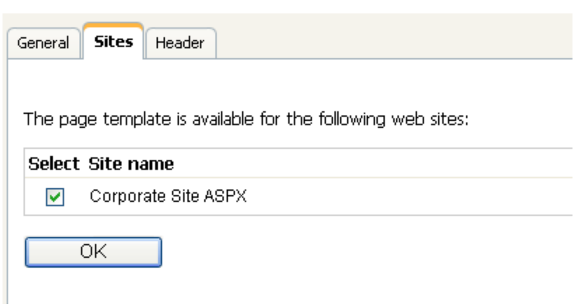
Click **OK**. Now enter the following value in the **File name** field:

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

It is the virtual path of our ASPX page.

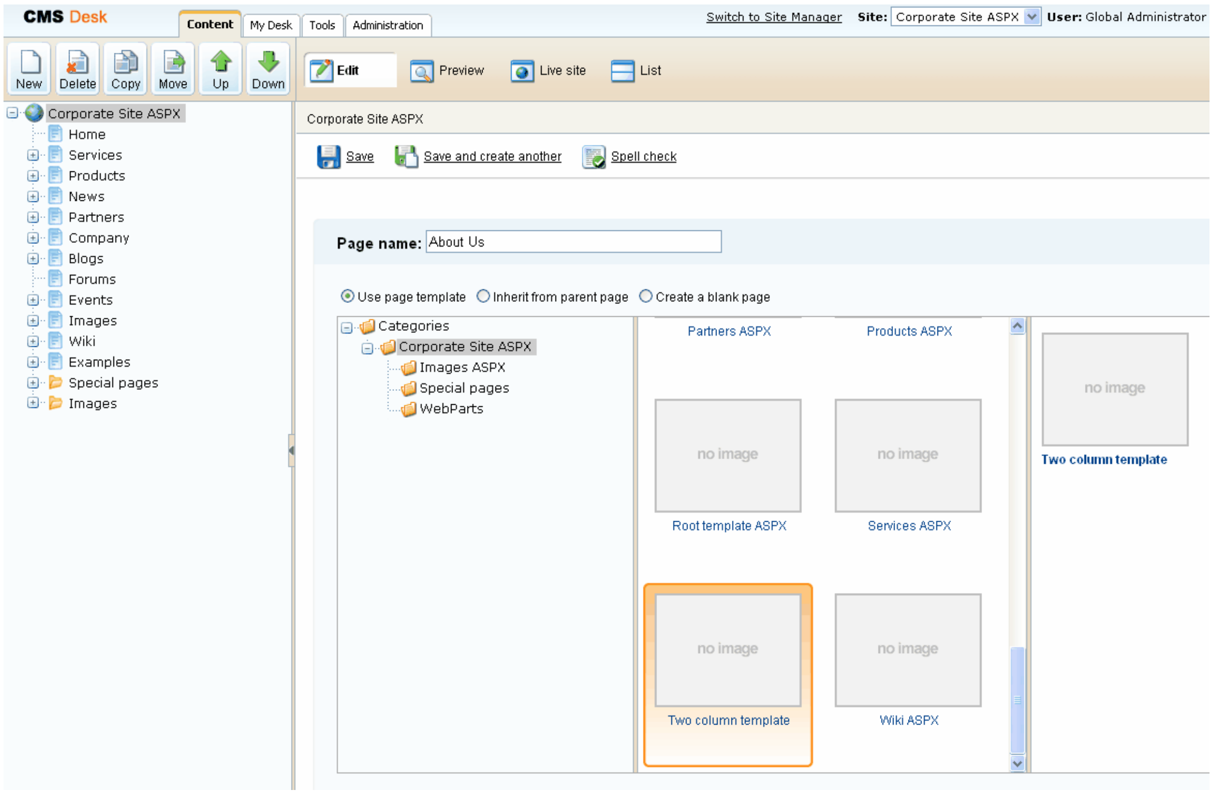



Click **OK** to save the changes. Now click the **Sites** tab and assign the page template to the current web site and click **OK**:

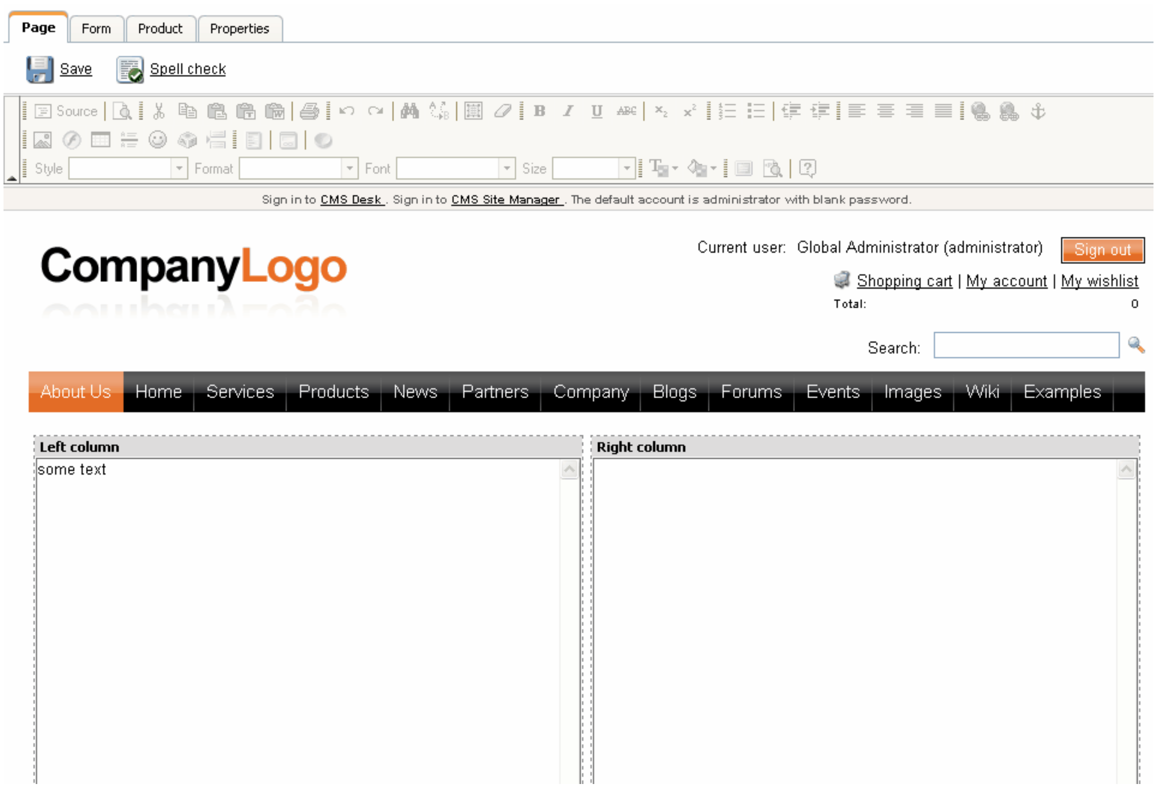


## Creating an About Us page based on the new page template

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



Click **Save** to create the new page. Click **Page** and you will see a page with editable regions like this:




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

## 11.3 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, that allows you to share the same site header and footer with logo, main menu, search box, etc. over all pages without having to create these sections on each page template again and again.

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

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

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

The following code sample defines a very simple master page:

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

    <form id="form1" runat="server">

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

</html>
```

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

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

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

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

```
using CMS.UIControls;
```

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

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

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

**[C#]**

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

    this.PageManager = this.CMSPageManager1;
}

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

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

**[VB.NET]**

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

    Me.PageManager = Me.CMSPageManager1
End Sub

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

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

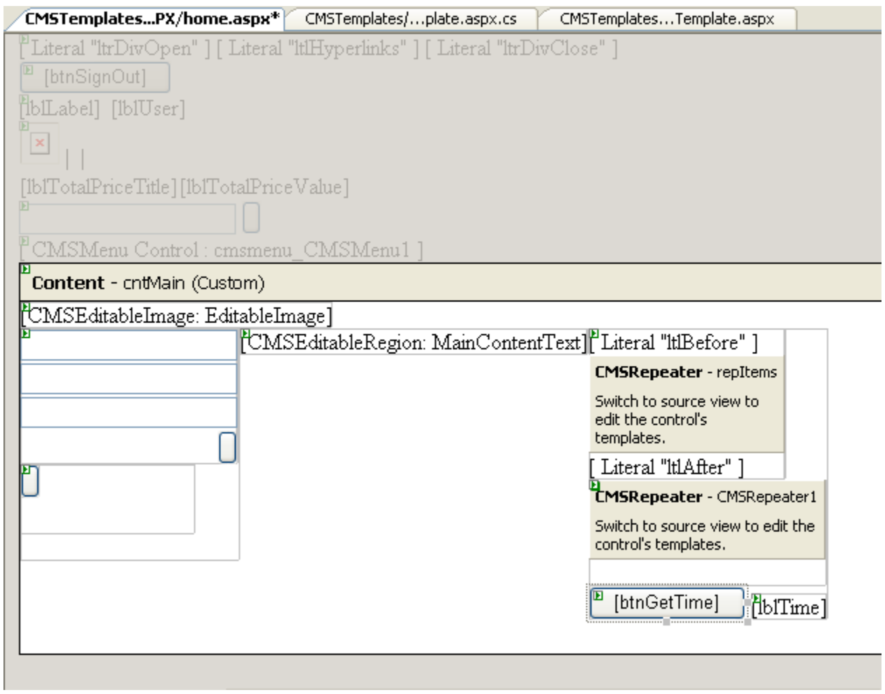
You should store master pages in the **CMSTemplates** folder together with page templates, so that they are exported with your web site.

## 11.4 Adding custom code to the ASPX page template

In this simple example, you will see how you can easily add custom code to the code of 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 web site **Corporate Site ASPX** for this example.

Open the web project in Visual Studio using the **WebProject.sln** file and open the page **home.aspx** located in folder **CMSTemplates\CorporateSiteASPX**.

Switch to the design mode 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.

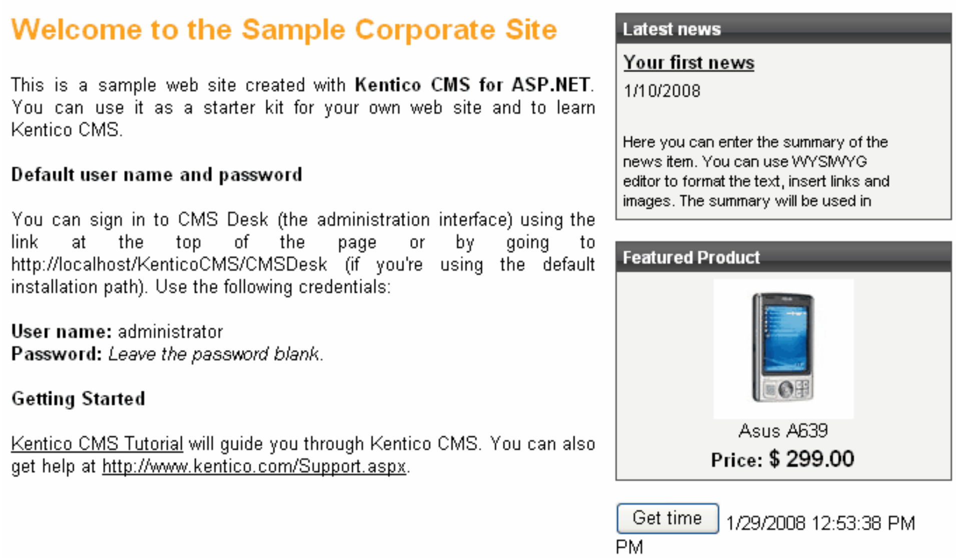



Double-click the button and add the following code inside the click event handler:

**[C#]**

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

Save all changes and see 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 standard debugging process in Visual Studio.

## 11.5 Combining ASPX templates and portal engine templates

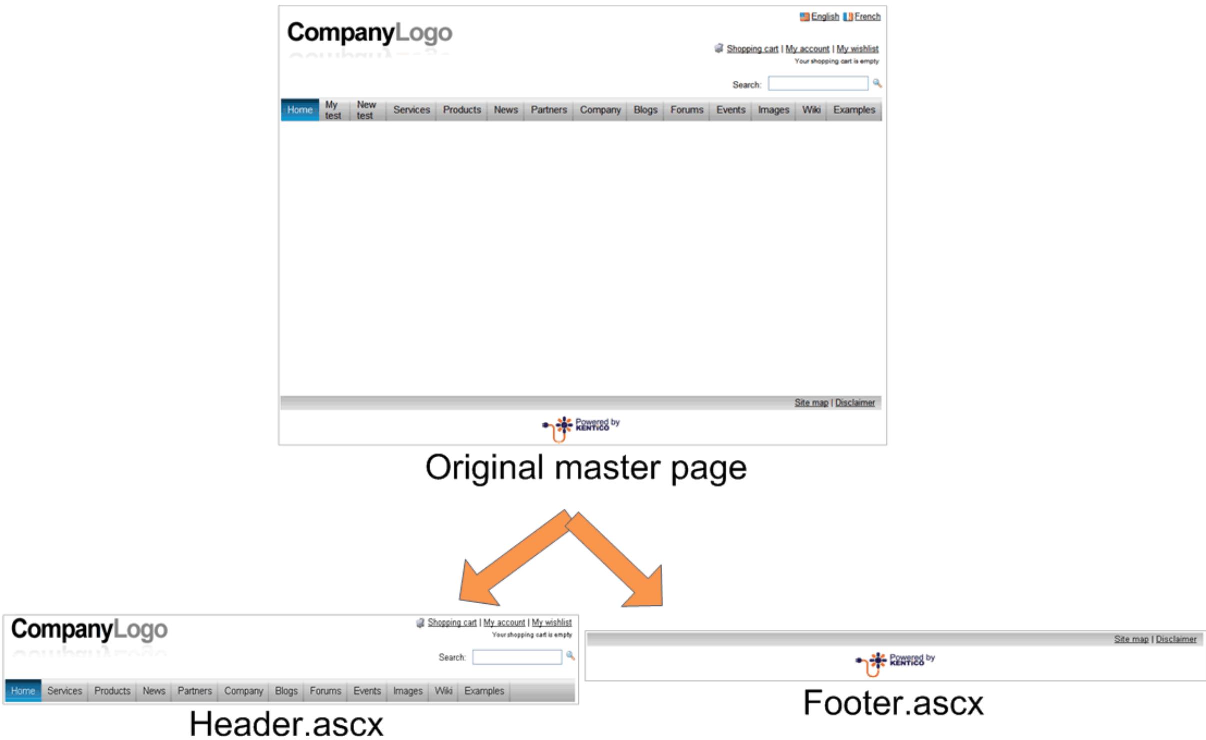
In some cases, you may need to combine ASPX page templates or external ASPX pages and the portal engine page templates. The following text describes how you can accomplish that.

You will create a web site using the portal engine model - it means you will create the master page and some of the pages using the portal engine page templates. The point here is how to "insert" the ASPX pages inside the same master page. Unfortunately, it's not possible to do that directly. However, you can use the following workaround:

**You will have two master pages - the portal engine master page defined in the root document in the content tree and the ASPX master page (.master file) used for all ASPX pages within web site.**

The drawback is that you will need to manage the master page in two places. But this issue also has a simple solution:

You need to 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, you take the web parts and HTML code "below" the Page placeholder and place it into the **Footer.ascx** user control as shown on the following figure:




Now, you delete the content of the original **portal engine master page** and place only the following web parts into the master page:

- **General/User control** web part displaying the **Header.ascx** user control
- **Page placeholder** web part
- **General/User control** web part displaying 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 manage the header and footer in a single place for both portal engine and ASPX master page.

# 11.6 Integration with your existing ASP.NET application

If you need to integrate some existing ASP.NET application with Kentico CMS web site, 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 three options 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 CMS and your application. The easiest way is to use Kentico CMS web project as the main project since it's already correctly configured for CMS and add your own ASPX pages and other files to this project.

If you need display your own ASPX pages inside the context of the web site, you can simply register them as page templates and then create new pages based on these page templates in the standard web site 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 web site.

If you wish to use a web site built using the Kentico CMS portal engine development model, please read also the chapter Combining ASPX templates and portal engine templates.

**2. Having CMS in the root and your application in a subfolder**

If your application can or needs to run separately from the CMS and you want the CMS to manage the main web site, you can place Kentico CMS web project in the root of the web site and your application into a subfolder. You will need to create a virtual directory for the subfolder so that your application runs correctly.

**3. Having your application in the root and CMS in a subfolder**

If your application can or needs to run separately from the CMS and your application is the main part of the web site and you wish to use the CMS only for some sub-section of the web site where you publish the content, you need to place the CMS into a sub-folder and create a virtual directory for it.

## Interaction between CMS and your application

If you need to include your application inside the web site (front-end), you can do that either through ASPX pages (see paragraph **Mixing both together** above) or you need to create ASCX user controls that you place into the CMS web site.

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 chapter Custom modules.

## Sharing security information between CMS and your application (single-sign-on)

If you wish to use single-sign-one for both your application and the CMS, you need to configure your environment as described in chapter Single sign-on. If you with to use a single system of permissions (authorization), you can leverage the permission system for custom modules as described in chapter Custom modules.

## 11.7 Displaying data from external database

Beside displaying Kentico CMS content, you can also display data from your external databases 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 if you created the web site from scratch.

### Example: Retrieving data from the sample Northwind database

In this simple example, you will see how to display data from the Categories table of the Northwind database using ADO.NET, in the sample Corporate Site ASPX web site. 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 2005 using the **WebProject.sln** file. Open the page CMSTemplates\CorporateSiteASPX\home.aspx.

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:

**[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. See the page on the live web site. You will see a grid with data from the external database:

| CategoryID | CategoryName | Description |
|---|---|---|
| 1 | Beverages | Soft drinks, coffees, teas, beers, and ales |
| 2 | Condiments | Sweet and savory sauces, relishes, spreads, and seasonings |
| 3 | Confections | Desserts, candies, and sweet breads |
| 4 | Dairy Products | Cheeses |
| 5 | Grains/Cereals | Breads, crackers, pasta, and cereal |
| 6 | Meat/Poultry | Prepared meats |
| 7 | Produce | Dried fruit and bean curd |
| 8 | Seafood | Seaweed and fish |

As you can see we used standard ASP.NET methods to display external data on the web site.

# Part XII

# 12 CSS stylesheets and design

## 12.1 CSS Overview

CSS stylesheets allow you to modify the design of the web site. You can use standard CSS styles with Kentico CMS as you are used to.

Every **web site** has some default CSS stylesheet. It can be modified in **Site Manager -> Sites -> Site Properties**, in field **Site CSS stylesheet**.

Every page can either use the web site 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**.

### Managing stylesheets

You can manage CSS stylesheets in **Site Manager -> Development -> CSS stylesheets**. Every stylesheet has the following properties:

| | |
|---|---|
| Stylesheet display name | The name of the stylesheet displayed to the users. |
| Stylesheet code name | The name of the stylesheet used in the code. |
| Stylesheet text | Standard CSS stylesheet code. |

You also need to enable the stylesheet for particular web sites on the **Sites** tab.

**Stylesheet URL**

You can receive any stylesheet using the URL like this:
**`web.com/CMSPages/GetCSS.aspx?stylesheetname=mystylesheetcodename`**

### Combining web site and page stylesheet

If you need to only combine the web site and page stylesheet, you need to configure the page for using its own style sheet and import the web site stylesheet using the following CSS directive:

```
@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:

## 12.2 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:

```
<cc1:CMSCalendar Runat="server">
<NextPrevStyle ForeColor="Red"></NextPrevStyle>
<WeekendDayStyle BackColor="#E0E0E0"></WeekendDayStyle>
</cc1:CMSCalendar>
```

**Web site design files**

It's recommended that you store all images or Flash movies that are part of the web site design template in the application theme folder. This ensures that the files are exported together with web site when your deploying it to some other server.

## 12.3 Printer friendly CSS styles

This chapter explains how to use the printer friendly CSS styles on your web site. 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.

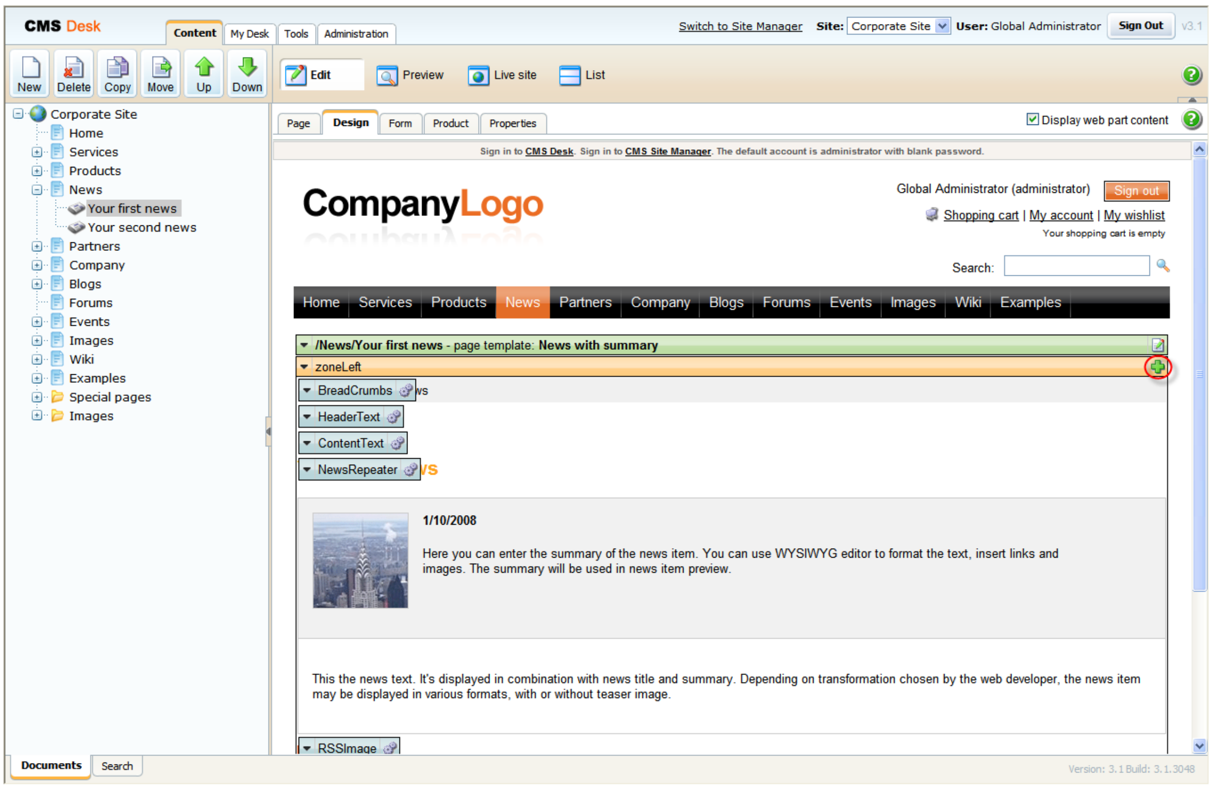
2. Add to the master page header tag link to appropriate CSS style. For example:

```
<link href="CMSPages/GetCSS.aspx?stylesheetname=Printer_styles" type="text/css"
rel="stylesheet" media="print" />
```

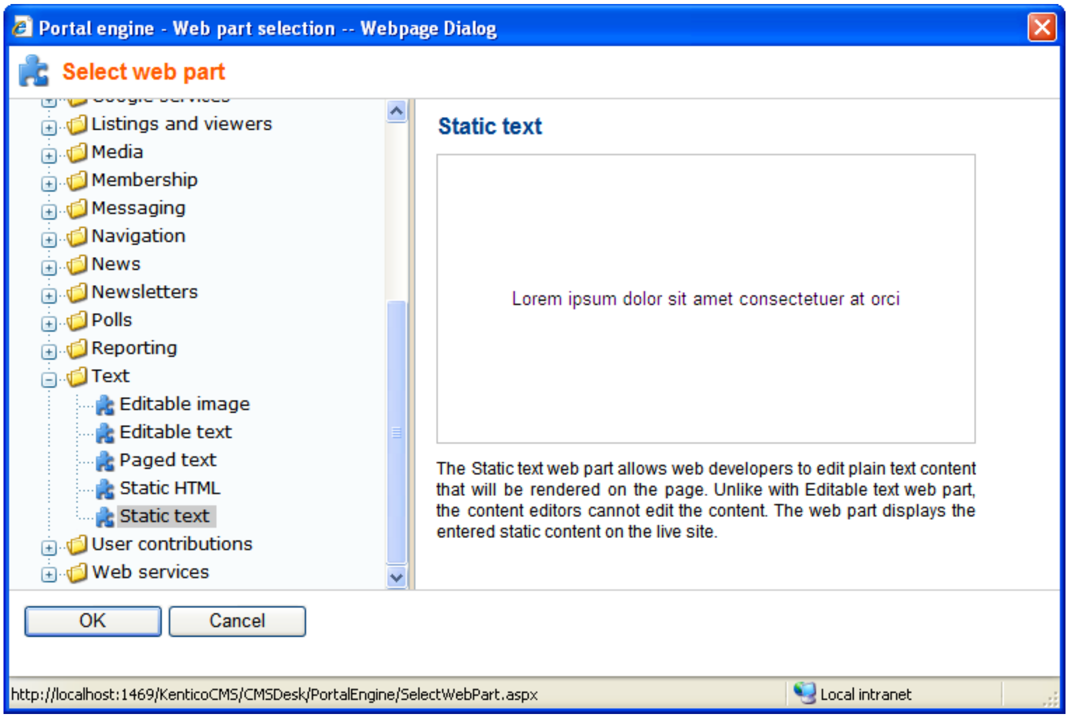
## 12.4 Print page

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 first news** -> **Design** and add click the **Add web part** button at the **zoneLeft**.

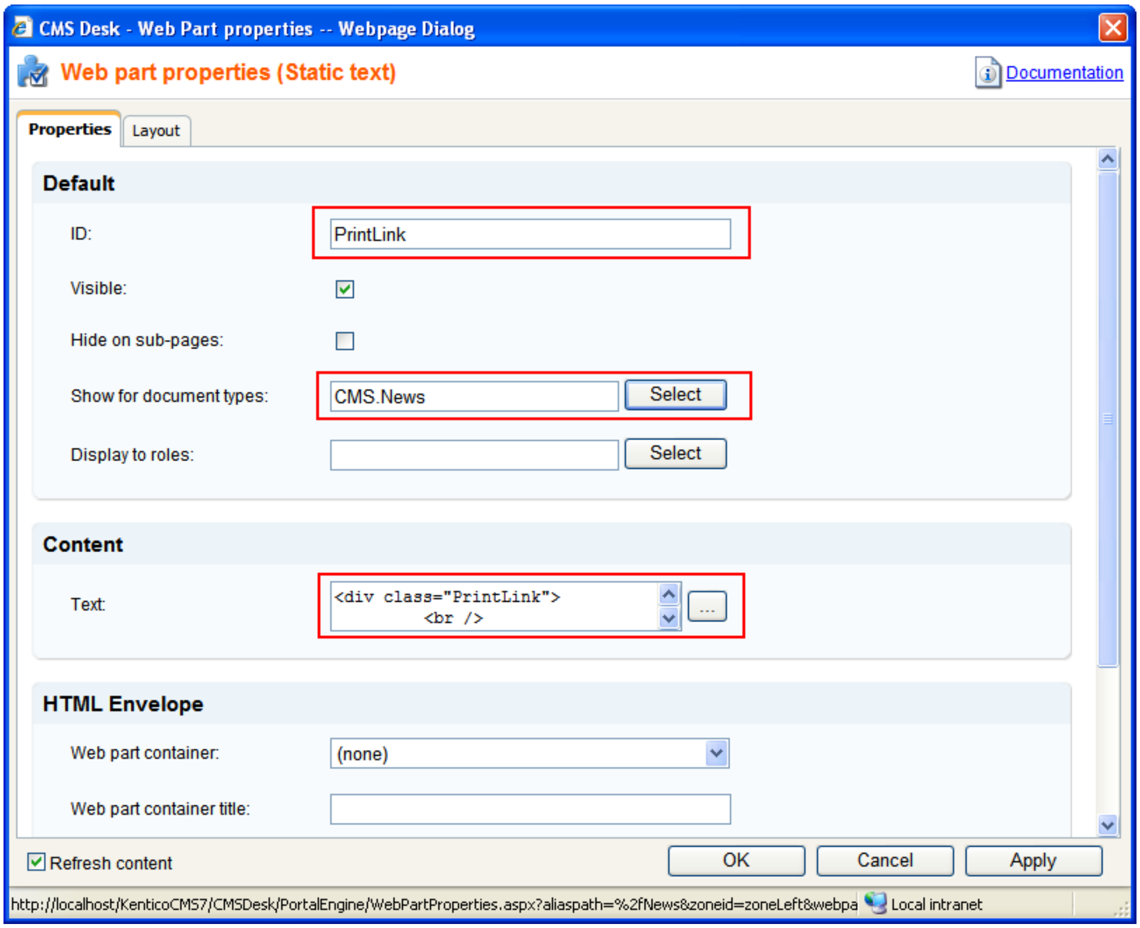


2) Select **Text**/**Static Text**.

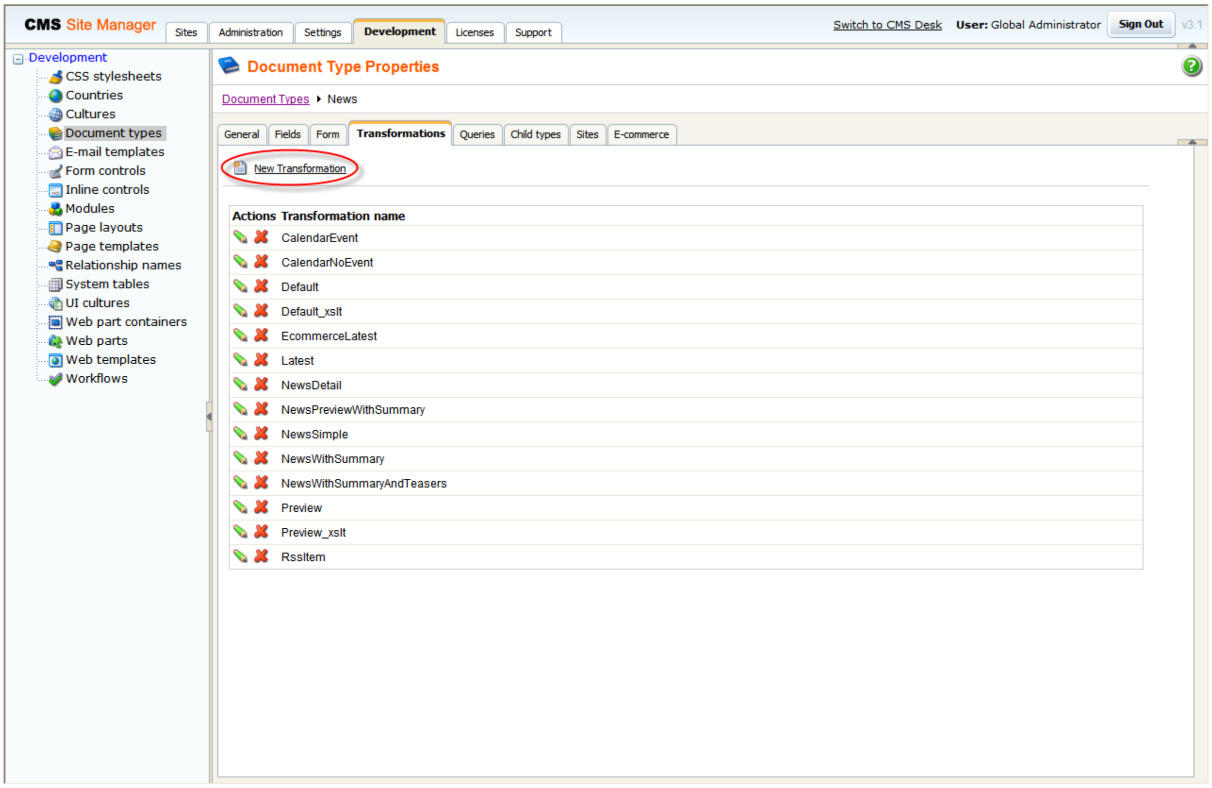


3) In the web part properties, enter *PrintLink* as **ID** and choose **CMS.News** for the Show for document types text box. Enter the following code into the **Text** text box. Then click **OK**.

```
<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>
```

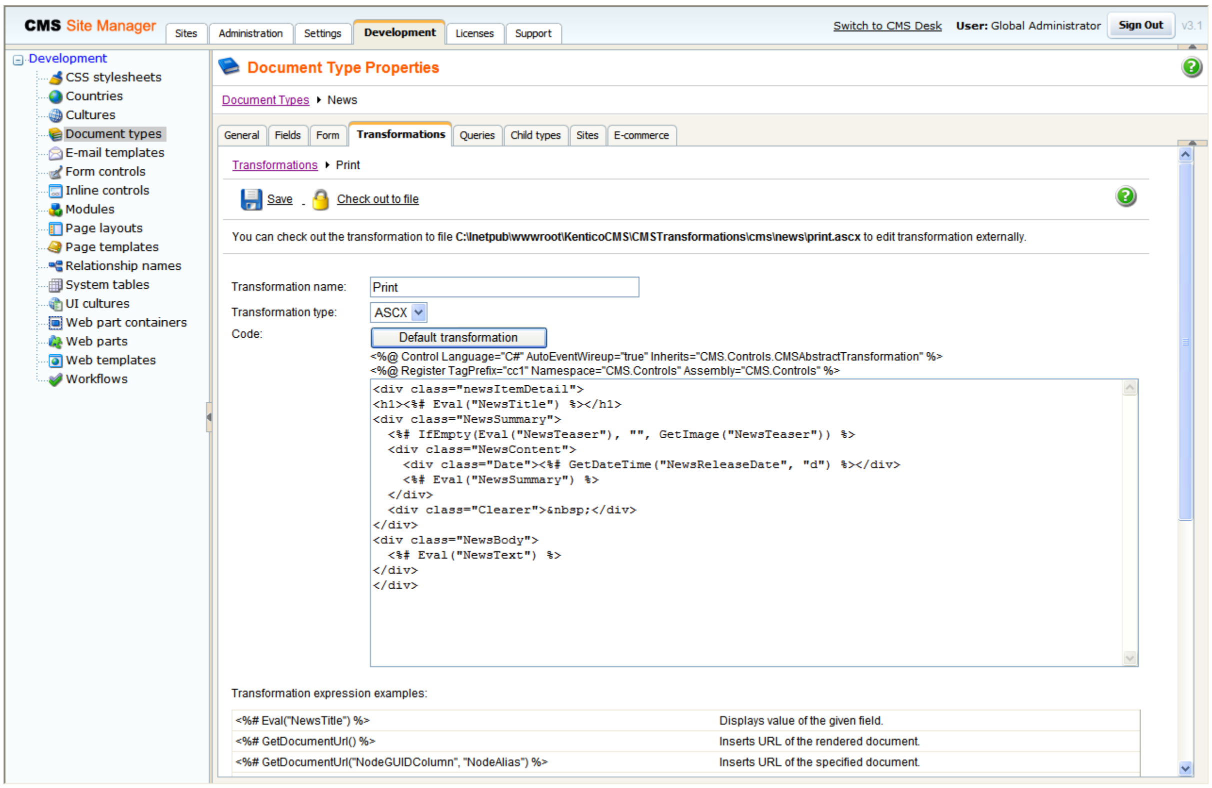


4) Now you have to specify the Print transformation for the new document type. Go to **CMS Site Manager** -> **Development** -> **Document Types** and click the **Edit** button next to the **News** document type. Switch to the **Transformation** tab and click **New Transformation**.

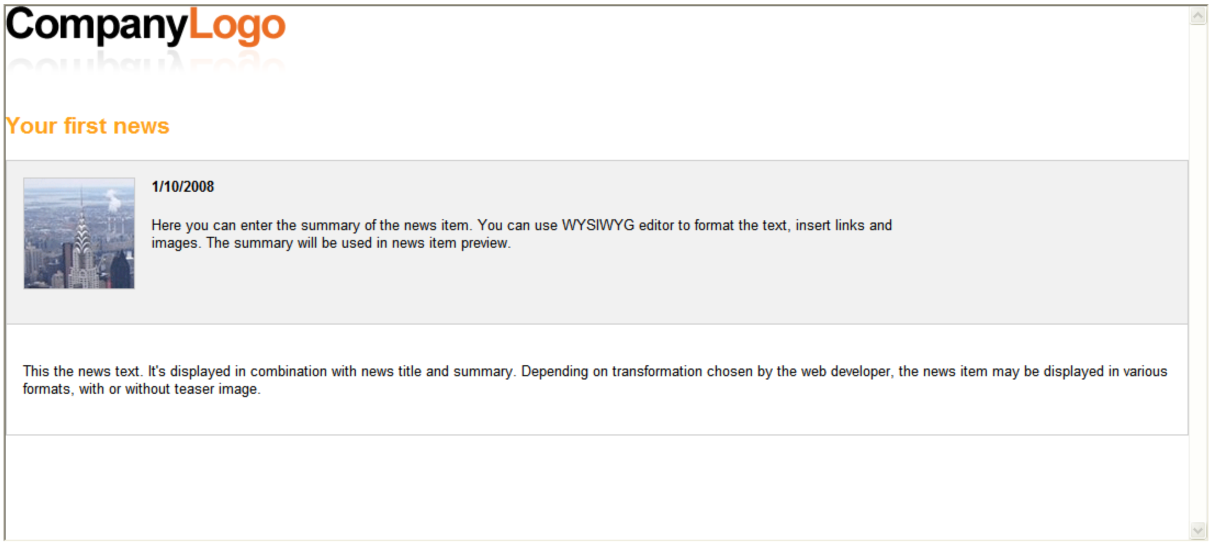


5) Enter *Print* as **Transformation name** and into the **Code** text box enter the following code. Then click **Save**.

```
<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>
```

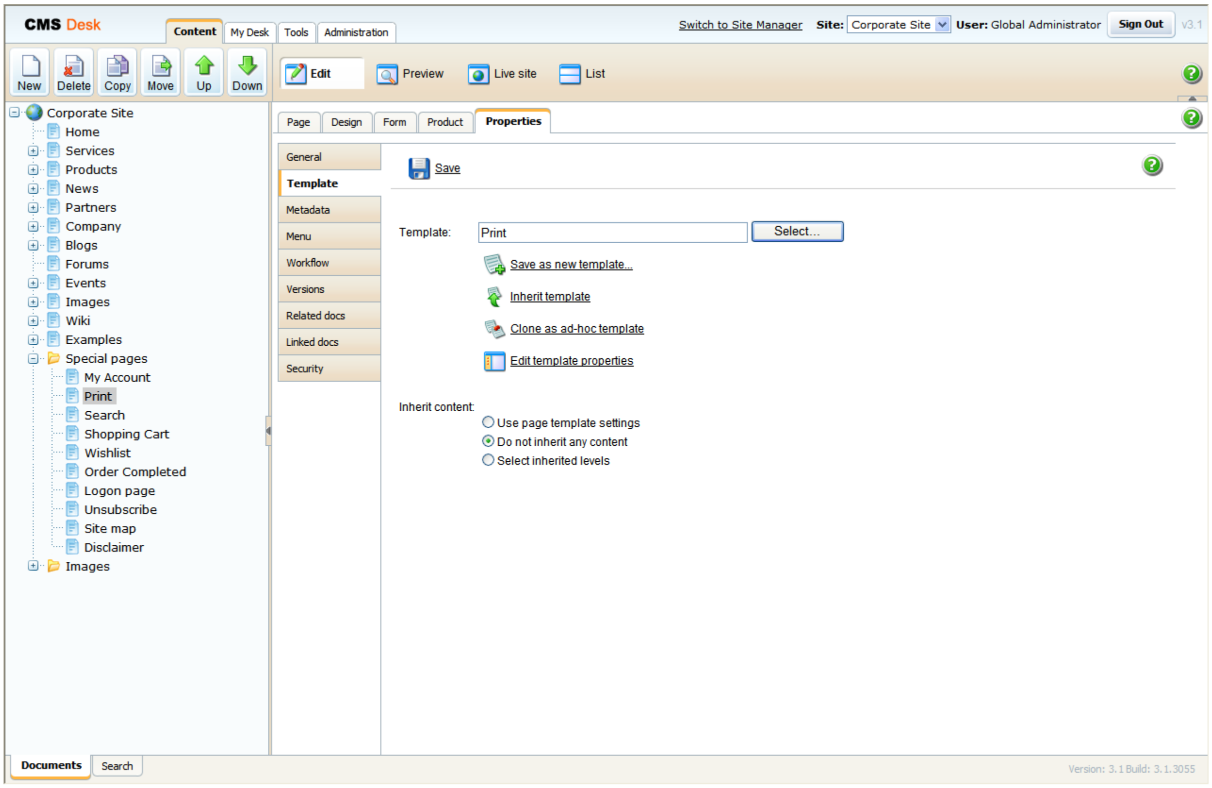


6) Now, go back to **CMS Desk** -> **Content** -> **News** -> **Your first news** and click the newly created **Print** button. You will be redirected to the **Print** page that displays the print version of the given news item.



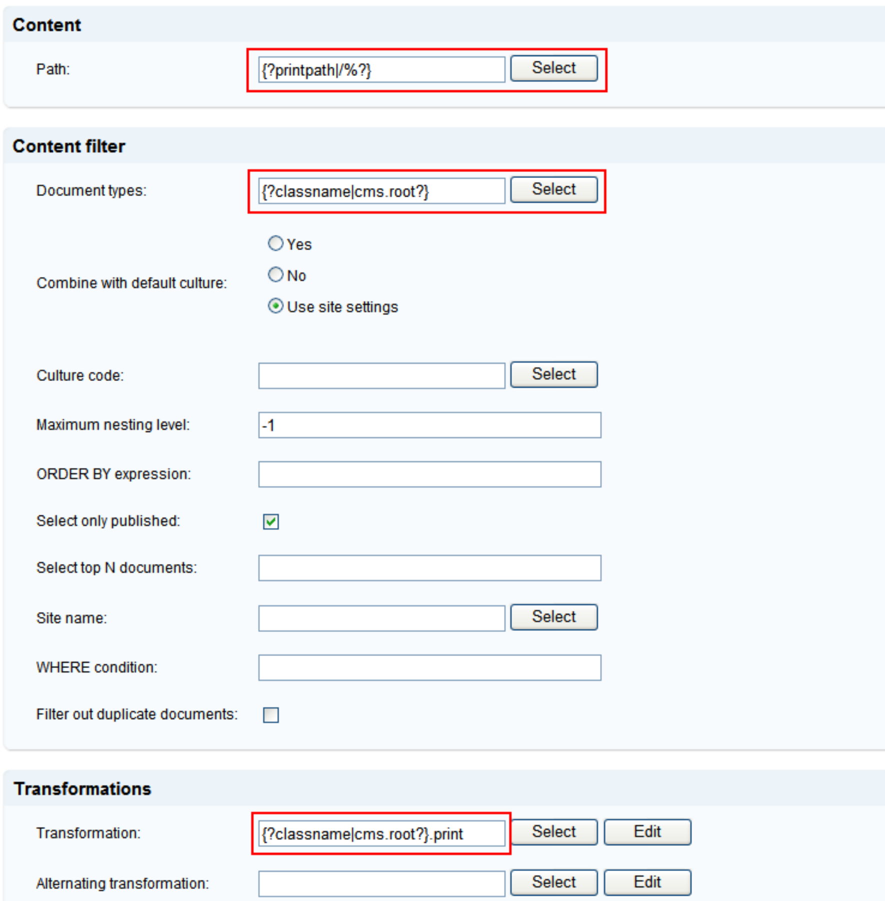

**Please note**:
For the sample Corporate Site, the **Print** page is already created. For your own web site, you have to create it by yourself.

Add a new blank page to the **Special pages** folder and name it **Print**. Firstly, you have to disable the content inheritance at **Properties** -> **Template**.



Every **Print** page contains the repeater web part that renders the **Print** transformation for the given document type that can be specified at **CMS 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

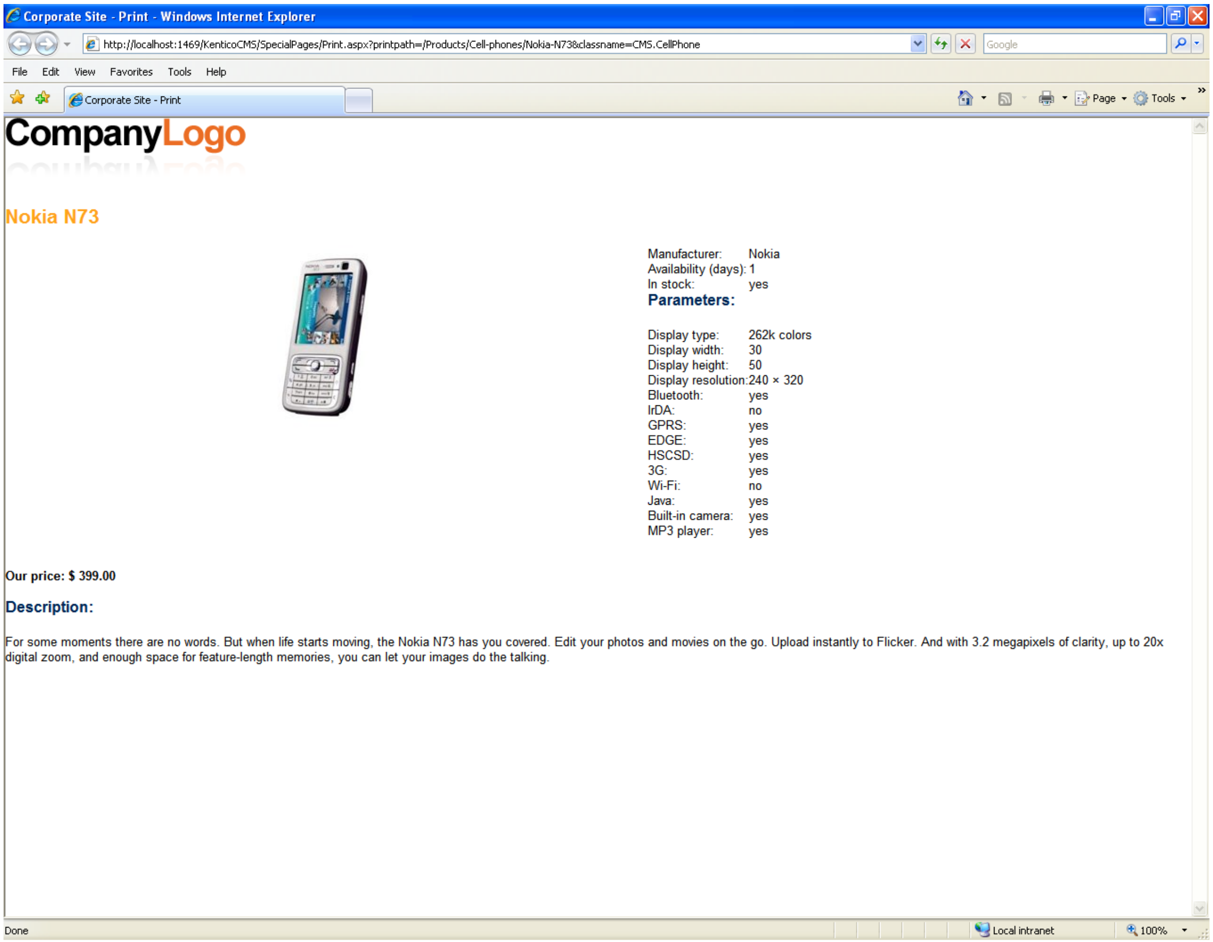



The **Path** text box specifies the path to the document whose print version you want to make. The value of the **printpath** macro expression is supplied in URL. If no value is supplied in URL (for instance you go directly on the **Print** page from the content tree and not through the print link button), 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.

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.


Corporate Site - Print - Windows Internet Explorer
http://localhost:1469/KenticoCMS/SpecialPages/Print.aspx?printpath=/Products/Cell-phones/Nokia-N73&classname=CMS.CellPhone
File Edit View Favorites Tools Help
Corporate Site - Print
Page
Tools
CompanyLogo
Nokia N73
Manufacturer: Nokia
Availability (days): 1
In stock: yes
Parameters:
Display type: 262k colors
Display width: 30
Display height: 50
Display resolution: 240 × 320
Bluetooth: yes
IrDA: no
GPRS: yes
EDGE: yes
HSCSD: yes
3G: yes
Wi-Fi: no
Java: yes
Built-in camera: yes
MP3 player: yes
Our price: $ 399.00
Description:
For some moments there are no words. But when life starts moving, the Nokia N73 has you covered. Edit your photos and movies on the go. Upload instantly to Flicker. And with 3.2 megapixels of clarity, up to 20x digital zoom, and enough space for feature-length memories, you can let your images do the talking.
Done
Local intranet
100%

# Part XIII

# 13 Document types and transformations

## 13.1 Document type overview

Each document 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.

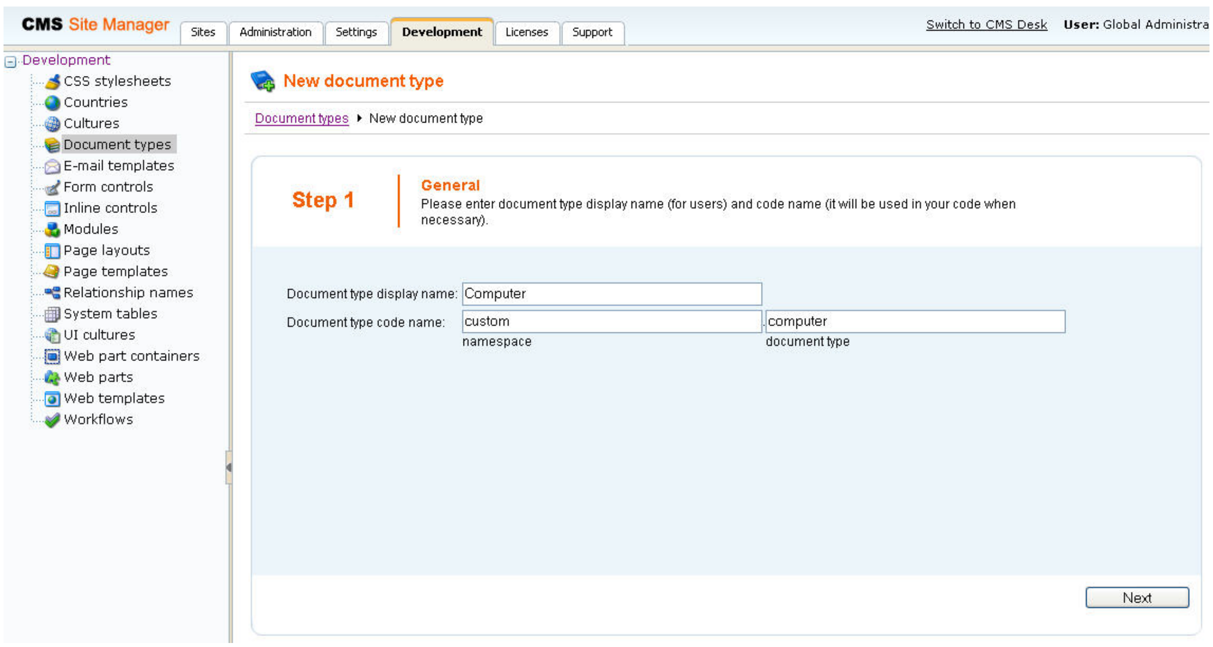
### General properties of a document type

| | |
|---|---|
| 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 the user must select some page template first when she creates 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 is the Page tab for such document type, 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. |

## 13.2 Defining a new document type

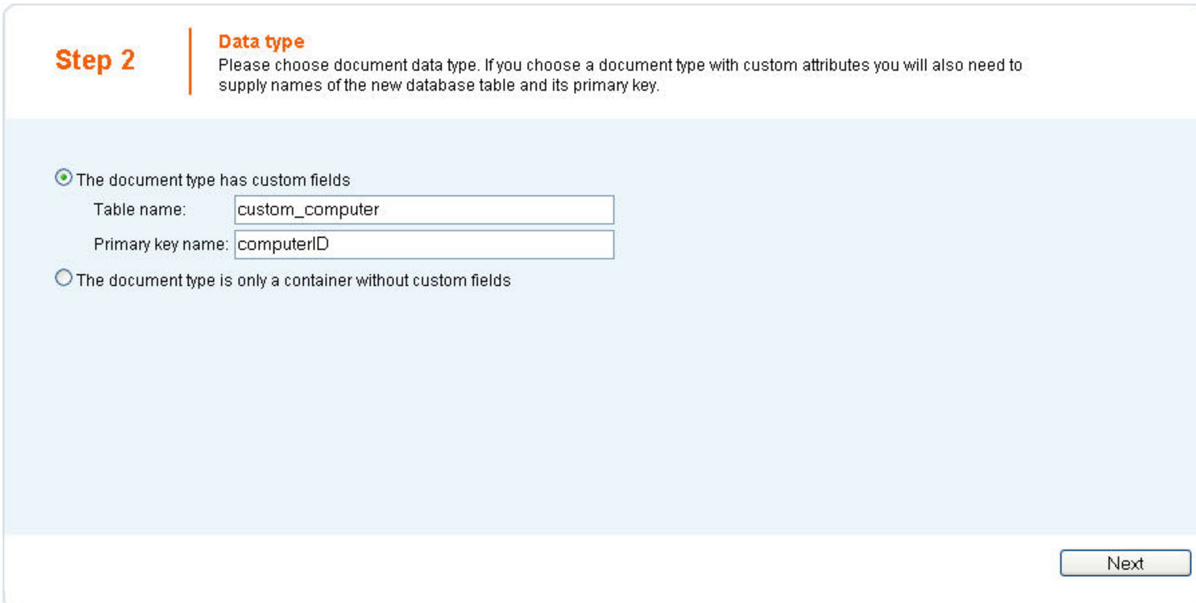
Go to **Site Manager -> Development -> Document types** and click **New document type**. You are redirected to the New document type wizard. In the first step, enter the following values:

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



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

- **Table name:** CUSTOM_Computer
- **Primary key name:** ComputerID




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

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

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

- **Attribute name:** ComputerProcessorType
- **Attribute type:** Text
- **Attribute size:** 200
- **Field caption:** Processor type
- **Field type:** Drop-down list, choose to use Options 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

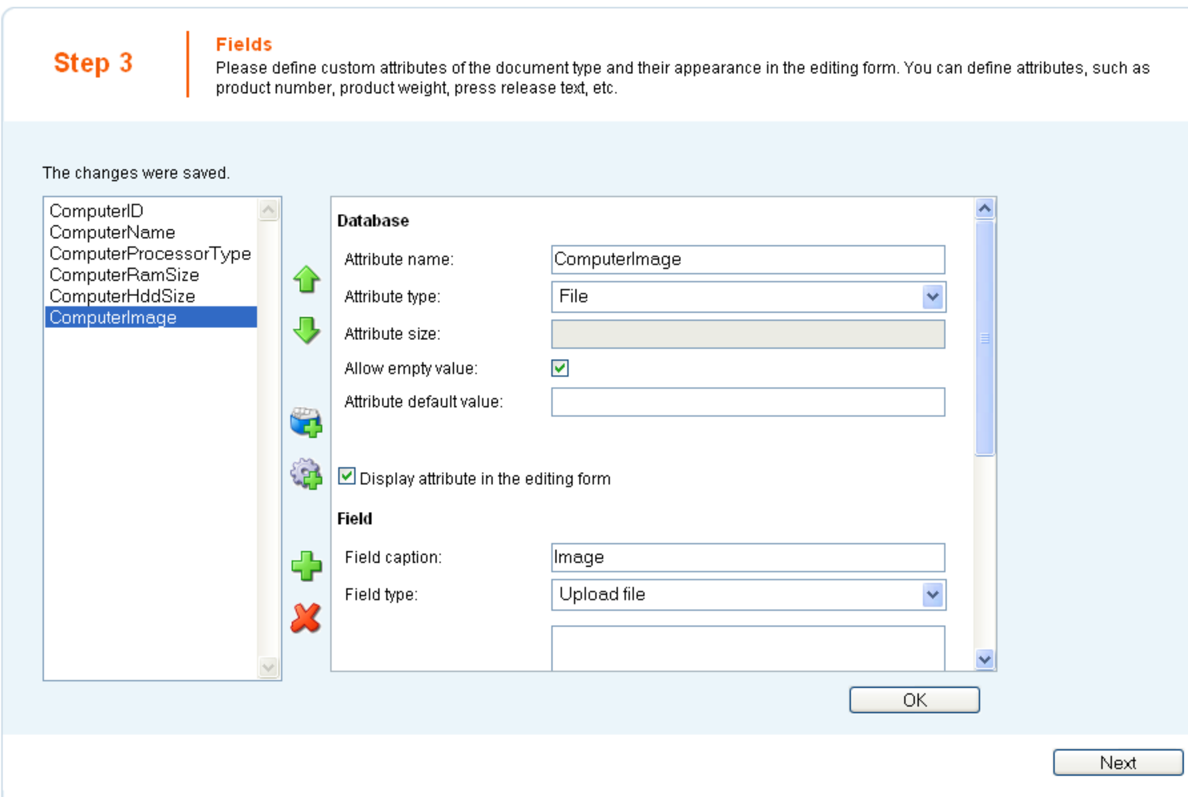
Click **OK**. Click **New attribute** (). Enter:

- **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

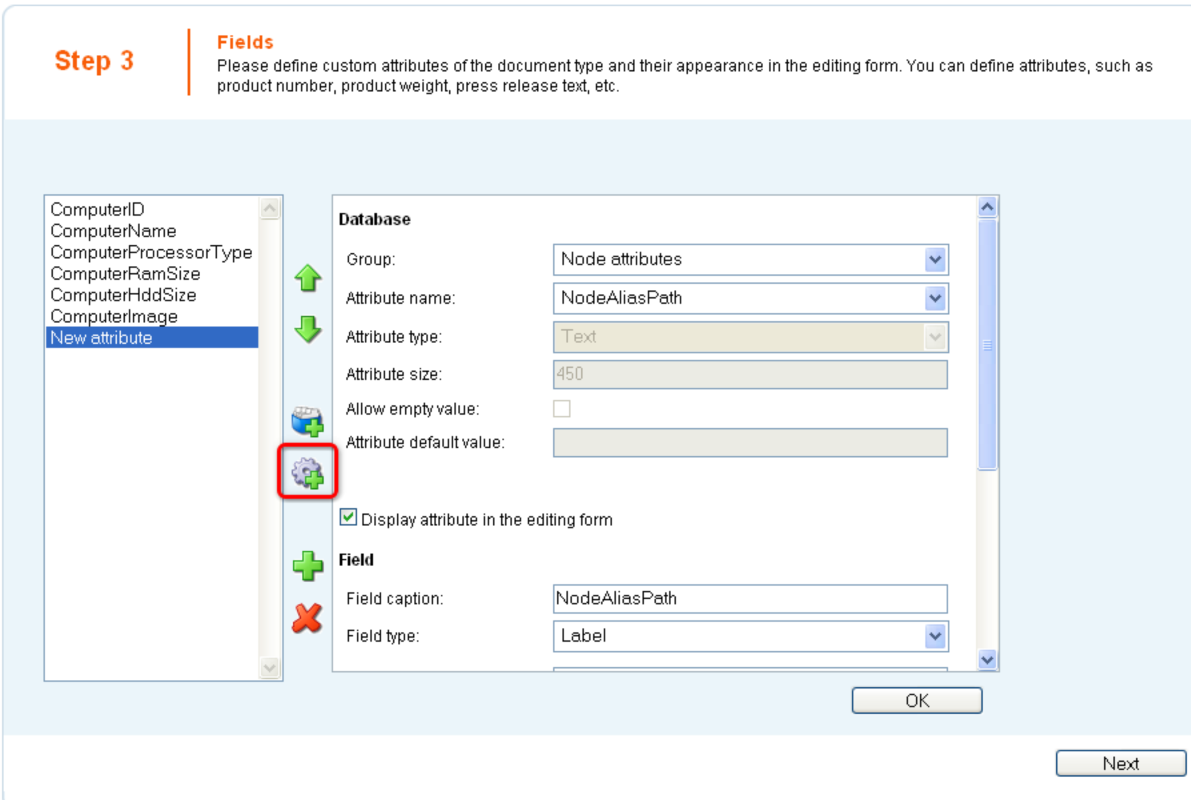
Click **OK**.



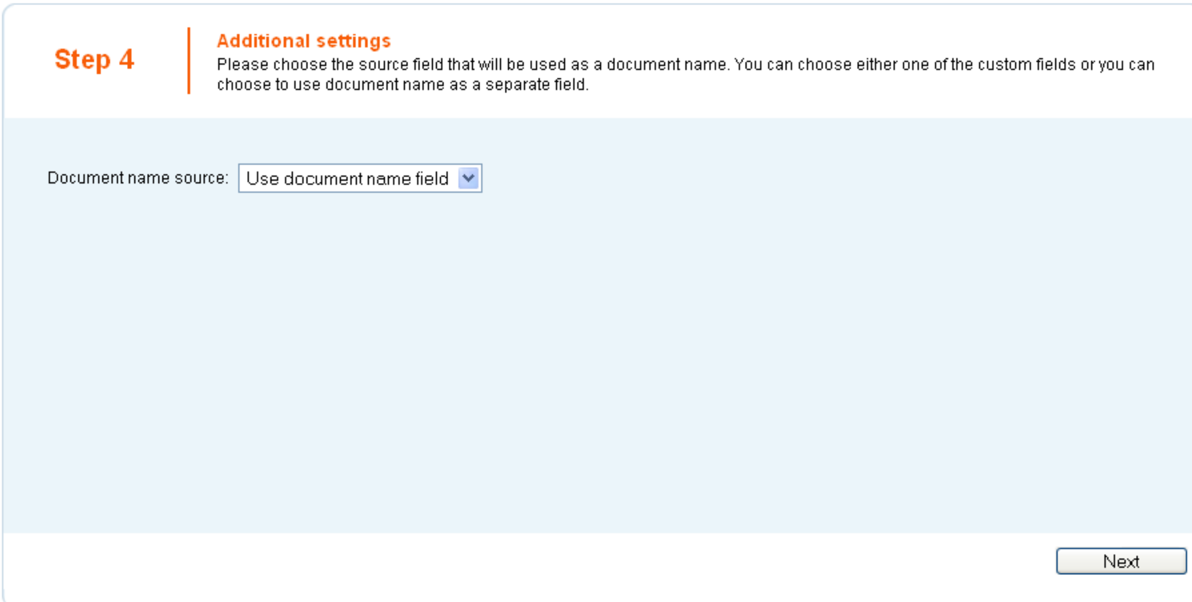
You can also define system fields to be displayed on the documents' **Form** tab. This can be done by clicking the **Add system attribute** icon. Using the **Group** drop-down list, you can then choose from the following two groups of system fields:

- **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.



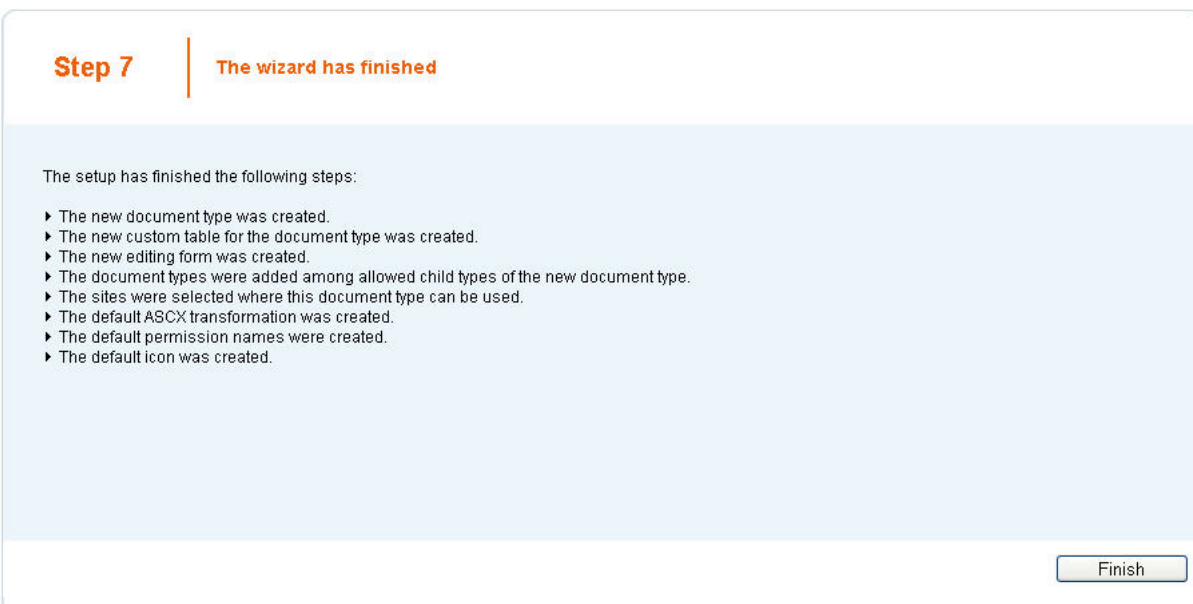
Click **Next**. Now you need to choose the field that will be used as document name. Choose the **Use document name field** option from the drop down list. It means that when you create a new computer document, its name will be automatically taken from the ComputerName value and this value will appear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




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

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

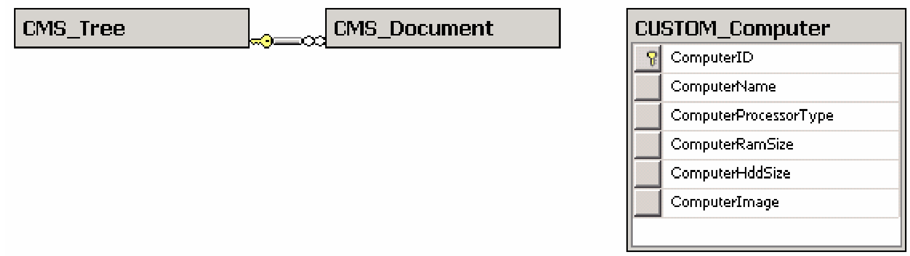

The wizard has finished the configuration of the new document type. It has automatically created not only the database table, but also the SQL queries for SELECT, INSERT, UPDATE, DELETE operations and a default transformation.




You have learned how to define a new document type.

**How the content is stored**

As you already know, the new document type Computer has its own database table. Each document is stored in three tables: CMS_TREE (tree structure), CMS_Document (document properties and metadata) and the custom table - in this case CUSTOM_Computer:




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

**Changing 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:

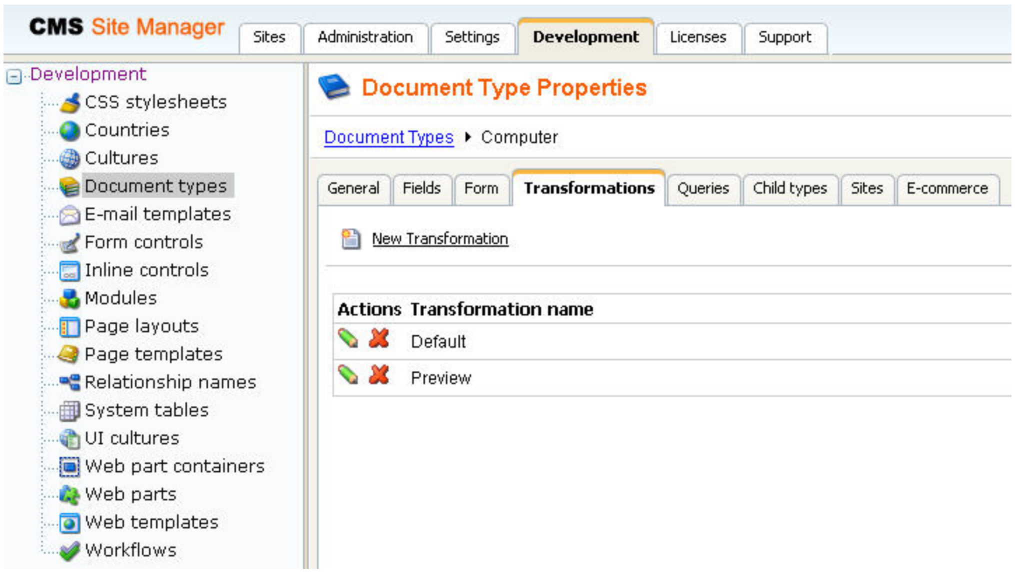
1) go to *<web project folder>\App_Themes\Default\Images\CMSDesk\Icons*
2) find your document type's icon named *<namespace>_<document type>.gif*
3) open the file in your image editor and modify it, or create a new image with the same name and format and replace the original one with it

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.

## 13.3 Writing transformations

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

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



As you can see, the wizard has created some default transformation. We will use it for our 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 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 directly under the transformation code.

We will create transformation for the list of computers. Now go back the transformation list and click **New transformation**. Enter the following values:

- **Transformation name:** preview
- **Transformation type:** ASCX (it's also possible to use XSLT, but we 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 (CMSXsltList)** web part. Otherwise it won't work.

**Transformations for multilingual web sites**

In some cases, you may need to display different text in transformations, based on the currently selected language. If you're using the built-in multilingual support, you can achieve this by creating another transformation with name ending with culture code.

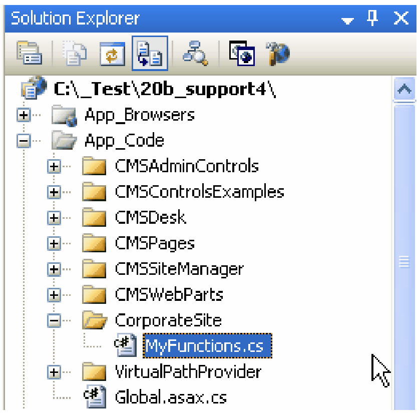
**Example:**
English (default language) transformation code name: cms.news.detail
French transformation code name: cms.news.detail_fr-fr

When you switch the content language to French, the French transformation will be automatically used in this case.

## 13.4 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 custom function that will return first N characters of the text and how to use it in a transformation.

Open the web project in Visual Studio 2005. Create a new folder under the **App_Code** section 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** (the 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 we can easily call it from our 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/>
```

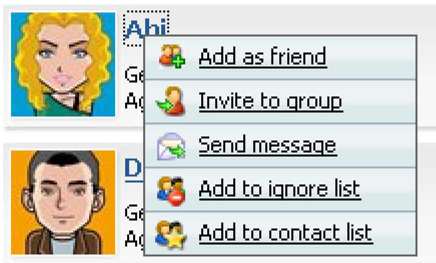
Click **OK** to save. Go to the live site and see the page with news listing. As you can see, the news summary text is truncated to first 10 characters.

You have learned how to write your own transformation methods.

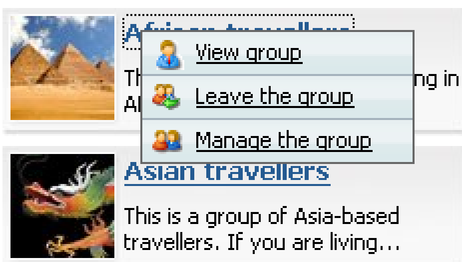
## 13.5 Context menus in transformations

When writing transformations for a web part displaying **Users** or **Groups**, you can enclose the transformation in a menu container control, which ensures displaying of context menus after right-click 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 enclose your transformation in the **cms: usermenucontainer**, resp. **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>
```

# Part XIV

# 14 Form controls

## 14.1 Form controls overview

Form controls allow you to use custom field types in editing forms that are based on Kentico FormEngine. These include:

- Document editing forms
- Web part properties
- BizForms
- Forms using system tables defined in Site manager -> Development -> System tables.

Form controls are standard ASCX controls that inherit from the **`CMS.FormEngine.FormEngineUserControl`** class.

### Managing form controls

You can register form controls in the catalog in **Site Manager -> Development -> Form controls**. The form controls have the following properties:

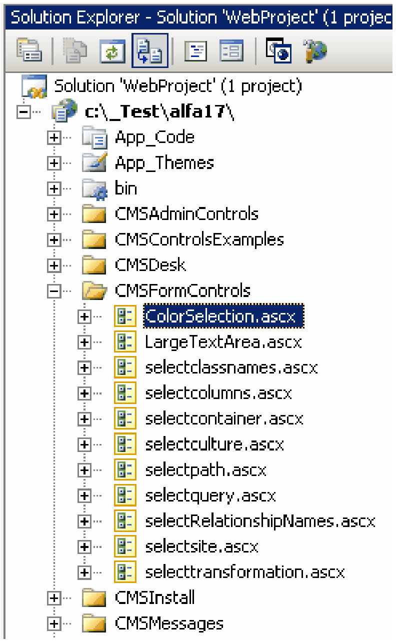
| | |
|---|---|
| Control display name | The name of the control displayed to the users. |
| Control code name | The name of the control used in the code. |
| Control file name | Relative path to the ASCX control starting from **`/CMSFormControls`** folder.<br><br>Example:<br>countryselection.ascx |
| 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 file (attachments). |
| Show control in BizForms | Indicates if the control should be offered among field types in the simplified mode in BizForms module. If you enable this box, you also need to enter the following values:<br><br>• **Default data type** - the type of the field that will be used by default when the user chooses to create a new field with this field type.<br><br>• **Column size** - default size of the database column (applies only to Text data type). |
| Show control in document types | Indicates if the control should be offered among field types in the document type field editor. |
| Show control in system tables | Indicates if the control should be offered among field types in the system table field editor. |
| Show control in web parts | Indicates if the control should be offered among field types in the web part property editor. |

## 14.2 Developing form controls

The following example shows how to create a form control that will allow uses to choose color from a dropdownlist.

1. Open the web project in Visual Studio 2005 (or Visual Web Developer) using the **WebProject.sln** file or using **File -> Open -> Web site** in Visual Studio.

2. Right-click the **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 in Design mode. Drag and drop a **DropDownList** control onto the form:


4. Switch to the code behind.

5. Change the following line

**[C#]**

```
public partial class CMSFormControls_ColorSelection : System.Web.UI.UserControl

to

public partial class CMSFormControls_ColorSelection : CMS.FormEngine.FormEngineUserControl
```

**[VB.NET]**

```
Partial class CMSFormControls_ColorSelection Inherits SystemWeb.UI.UserControl
```

to

```
Partial class CMSFormControls_ColorSelection Inherits CMS.FormEngine.FormEngineUserControl
```

It ensures that our user control inherits from the FormEngineUserControl class and can use its standardized properties.

6. Modify the code behind like this:

**[C#]**

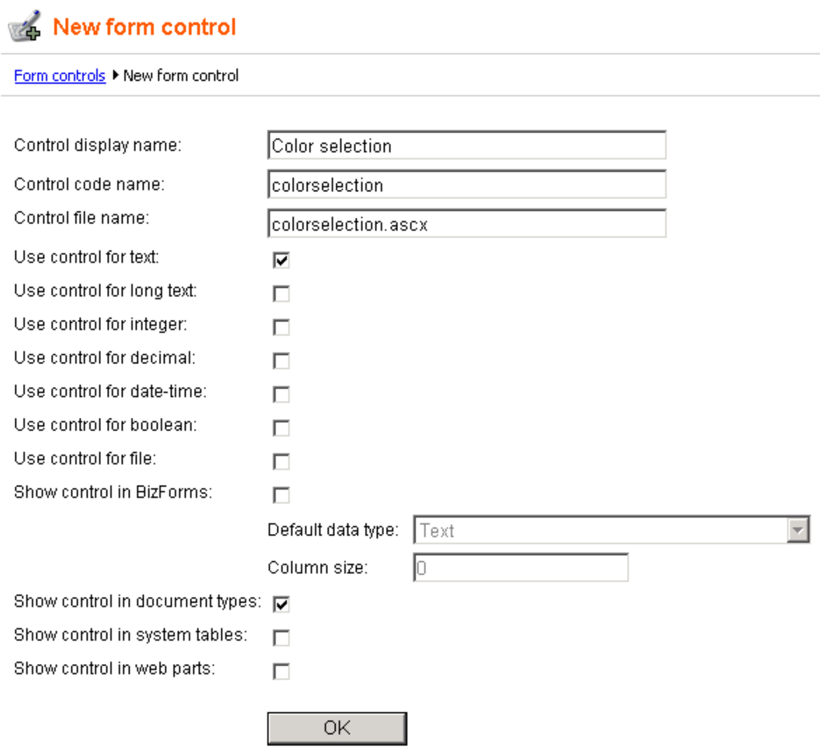
```
protected void Page_Load(object sender, EventArgs e)
{
     // Ensure drop down list options
     EnsureItems();
}

/// <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 no, 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 no, 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
```

7. 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

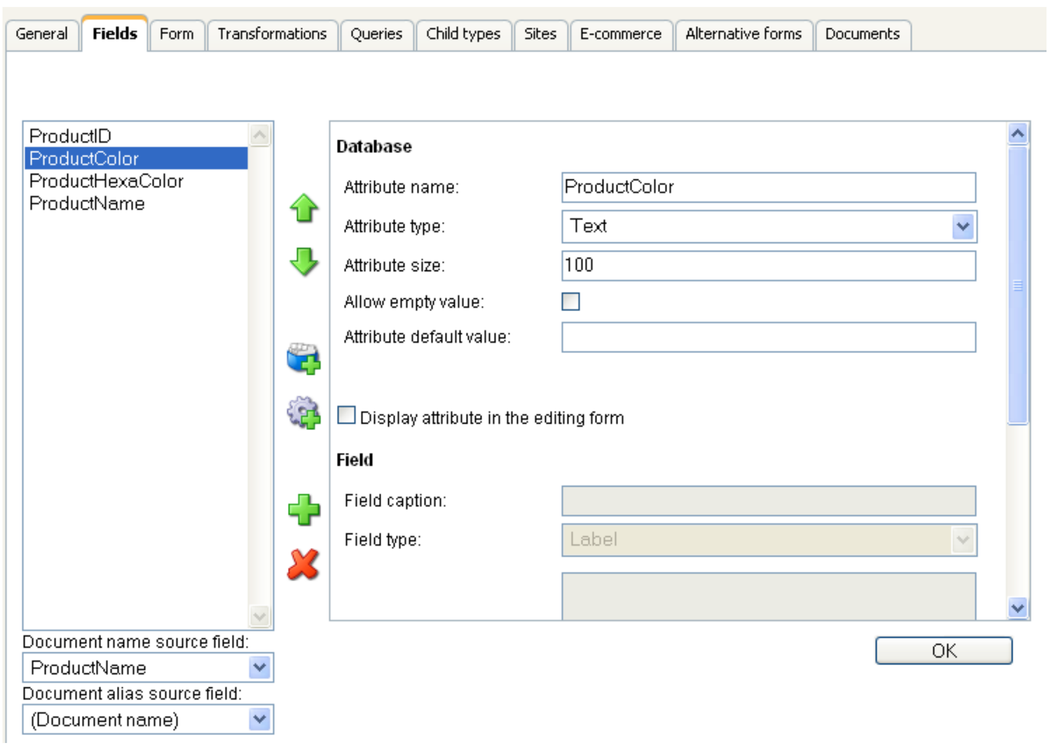
   Check the **Use control for text** box, check the **Show control in document types** box and click **OK**.



8. 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 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



9. 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 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 its 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 its 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.

# Part XV

# 15 Inline controls

## 15.1 Inline controls overview

Inline controls are user controls (ASCX) that can be placed into text using a special expression in format **`%%control:MyUserControl%%`**
where MyUserControl is name of the user control file without the ".ascx" extension. The system then dynamically loads the controls when the page is diplayed on the live site.

The controls may contain any functionality, such "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 the inline controls into the text

The inline controls can be inserted into the text using the **Insert inline control** command in the WYSIWYG editor toolbar or (if the text is not edited by the WYSIWYG editor) they can insert the special expression **`%%control:MyUserControl%%`**

The inline control may also have a single parameter that is passed in the following format:
**`%%control:MyUserControl?myparametervalue%%`**

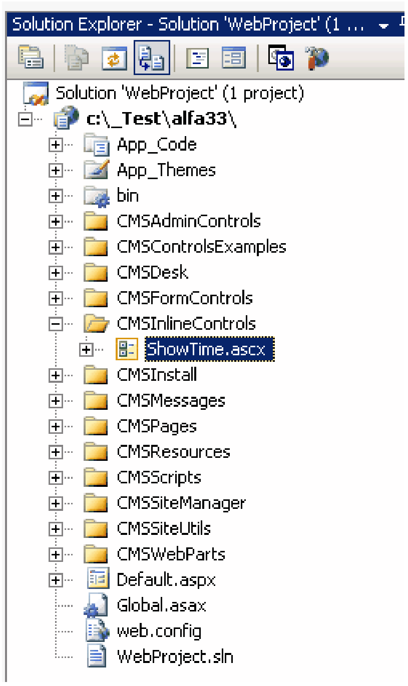
In the next chapter, you will learn how to develop your own inline controls.

## 15.2 How to develop inline controls

This chapter will show you an example of inline control development. We will create a simple control that will display the current time when the button is clicked.

1. Open the web project in Visual Studio 2005 (or Visual Web Developer) using the **WebProject.sln** file or using **File -> Open -> Web site** in Visual Studio.

2. Right-click the **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** mode and drag and drop a Label control and a Button control on the form:

4. Double-click the Button control and enter the following code into the Button1_Click method that will ensure displaying of current date and time in 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]**

(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
```

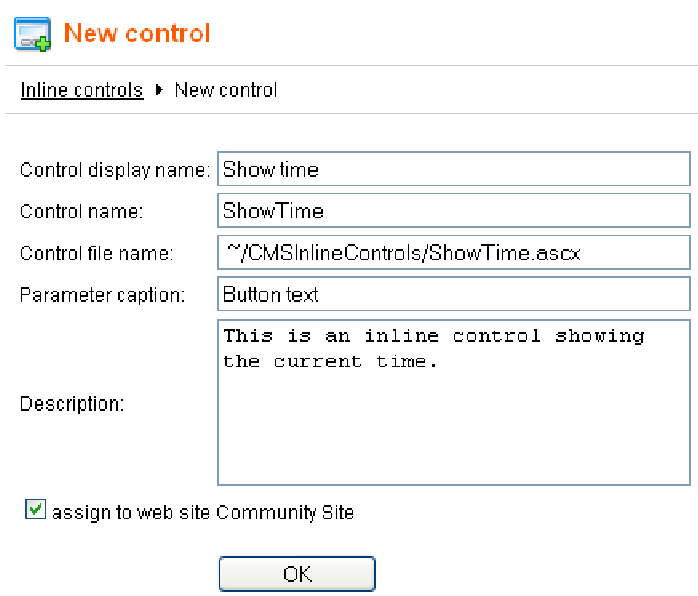
**What you did**

This code sets the Button1 caption based on the inline control parameter.

7. Save the changes and run the project.

8. Go to **CMS Site Manager -> Development -> Inline controls**. Click **New control** and enter in the following values:

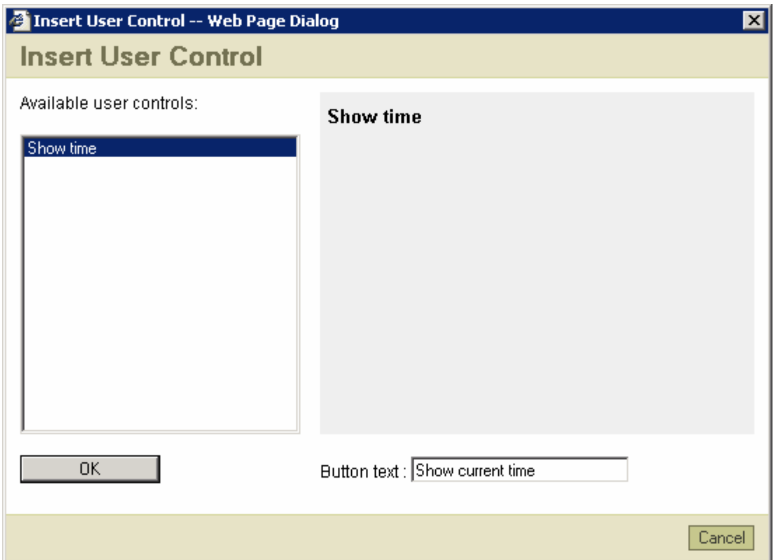
- **Control display name:** Show time
- **Control name:** showtime (the name of the user control without extension)
- **Control file name:** ~/CMSInlineControls/ShowTime.ascx
- **Parameter caption:** Button text



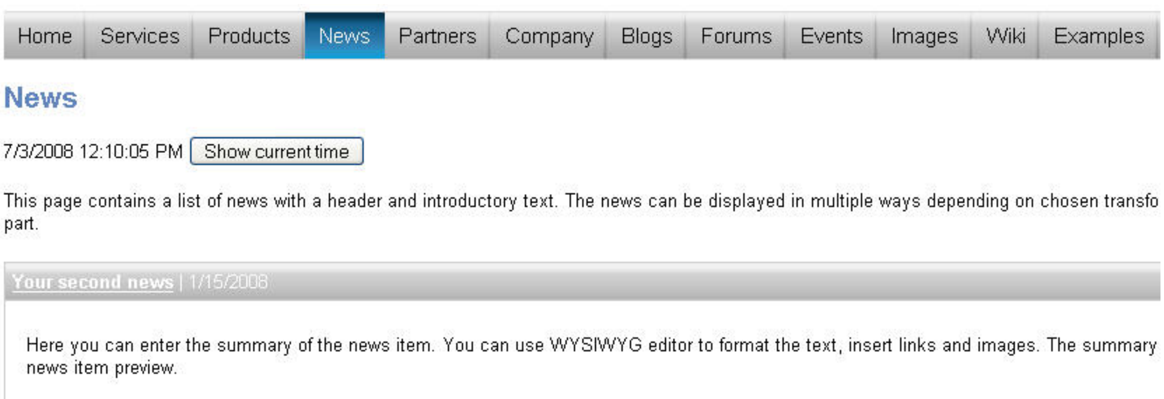

Click **OK**.

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 current time*. Click **OK**. The special expression is 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 current date and time:



# Part XVI

# 16 Web part development

## 16.1 Web part overview

Web parts represent a block of content or combination of content and functionality. They are the basic building block of page templates.

Web part is a user control (ASCX) that inherits from the **CMSAbstractWebPart** class. You can easily create your own web parts as described in chapter Developing web parts.

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

| | |
|---|---|
| Web part display name | The name of the site displayed to the users. |
| Web part code name | The name of the site used in the code. |
| Web part category | Here you can choose the category of the web part catalog where the web part is placed. |
| Web part file name | Contains a relative path to the user control that implements the web part. The path starts from the CMSWebParts folder. Example:<br><br>`Search/cmscompletesearchdialog.ascx`<br><br>It's recommended that you organize the web parts on the disk in the same way as in the categories. |
| Web part description | Description. |

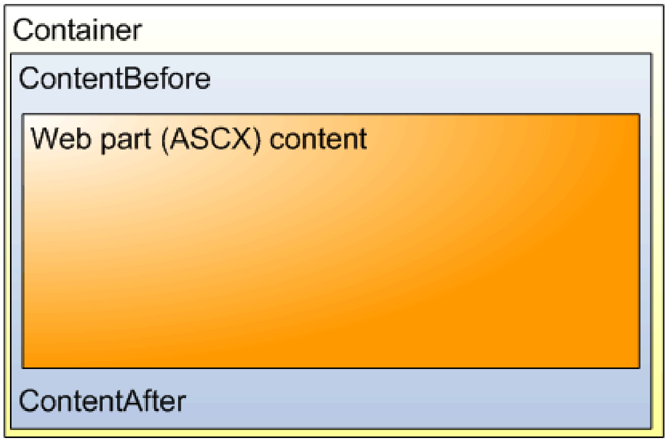
On the **Properties** tab, you can define the web part properties and how they appear in the **Web part properties** editing form. Each web part must have at least the following property:

| | |
|---|---|
| WebPartControlID | ID of the web part on the page. |

It's also recommended that you use the following properties:

| | |
|---|---|
| HideOnSubPages | Indicates if the web part should be hidden on sub-pages. |
| ContentBefore | HTML content displayed before the web part. You can use it to display some header or add some encapsulating code, such as <div> or <table> element for required layout. |
| ContentAfter | HTML content displayed after the web part. You can use it to close the tags contained in the ContentBefore value, such as </div> or </table> element for required layout. |
| Container | Container (box) displayed around the web part. This value contains the code name of the container defined in **Development -> Web part containers**. See also: Containers overview |
| ContainerTitle | Container title displayed if you specify the container. |

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

If your web part consists of several files (such as ASCX controls, images, js scripts, etc.), you should place these files in the subfolder under the folder where your main web part ASCX file is placed. If the 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 web site 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 a complete web part documentation and print it by going to **<web site URL>/CMSPages/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.

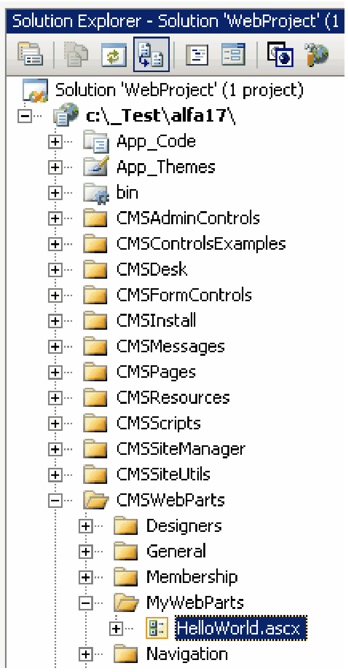
## 16.2 Developing web parts

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 in the label.

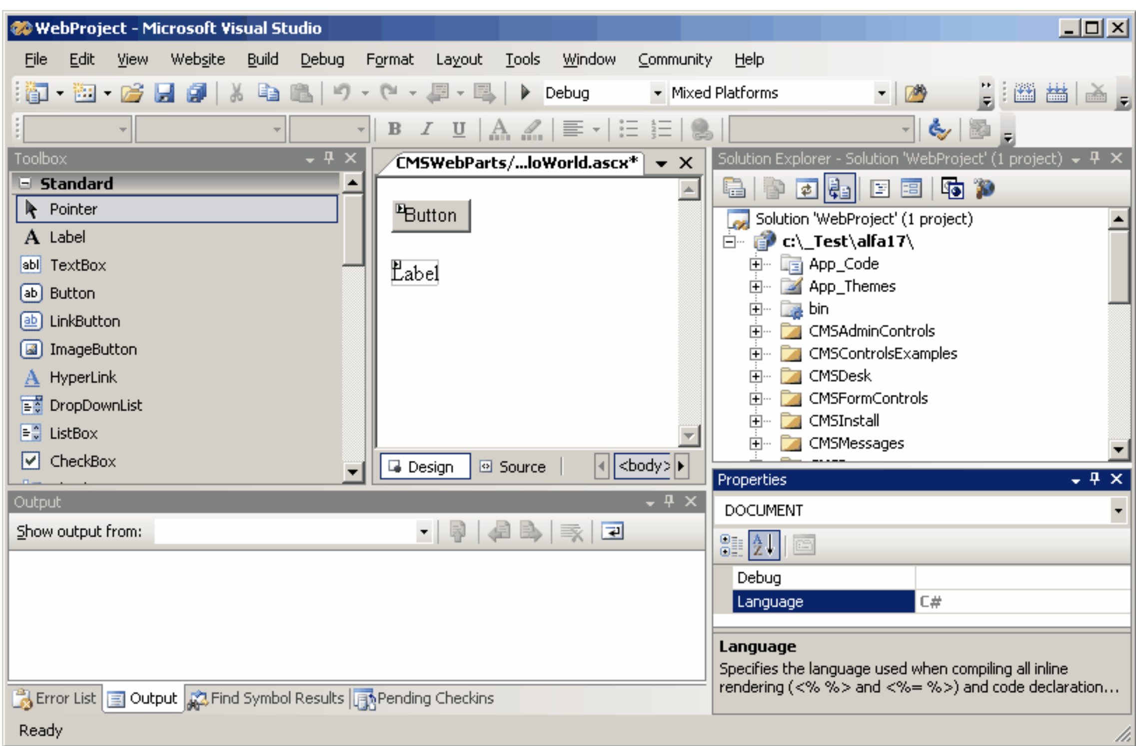
1. Open the web project in Visual Studio 2005 (or Visual Web Developer) using the **WebProject.sln** file or using **File -> Open -> Web site** in Visual Studio.

2. Right-click the **CMSWebParts** folder in the **Solution Explorer** window and choose **New Folder**. Rename the folder to **MyWebParts**.

3. Right-click the **MyWebParts** folder and choose **Add New Item**. Choose to create a new **Web User Control** and call it **HelloWorld.ascx**.



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



5. Double-click the Button control and add the following code to the Button1_Click method:

**[C#]**

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

**[VB.NET]**

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

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

**[C#]**

```
using CMS.PortalControls;
```

**[VB.NET]**

```
Imports CMS.PortalControls
```

7. Change the following line:

**[C#]**

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

to

public partial class CMSWebParts_MyWebParts_HelloWorld : CMSAbstractWebPart
```

**[VB.NET]**

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

to

Partial class CMSWebParts_MyWebParts_HelloWorld
    Inherits CMSAbstractWebPart
```

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

8. Add the following code to the Page_Load method:

**[C#]**

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

**[VB.NET]**

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

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

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

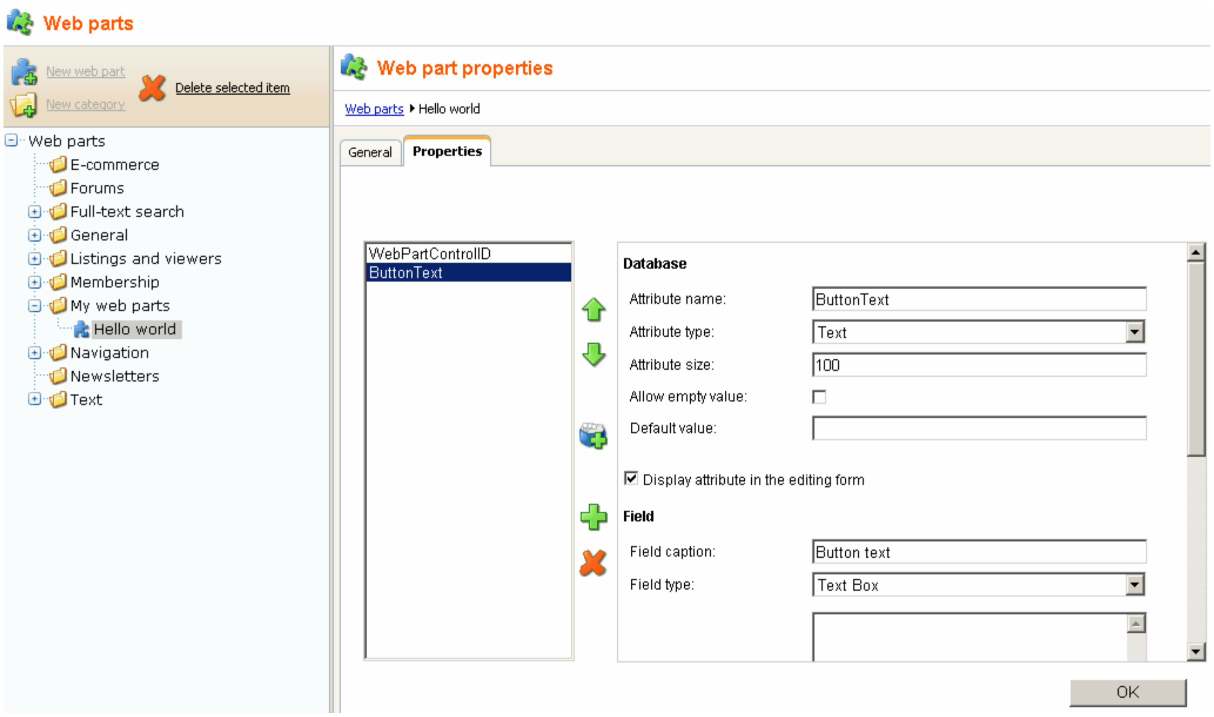
9. Save all changes.

10. Open **Site Manager -> Development -> Web parts**, click the root and click **New category**. Enter *My web parts* into the **Category 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:
    - Web part display name: Hello world
    - Web part code name: HelloWorld
    - Web part file name: MyWebParts/HelloWorld.ascx

    Click **OK**.

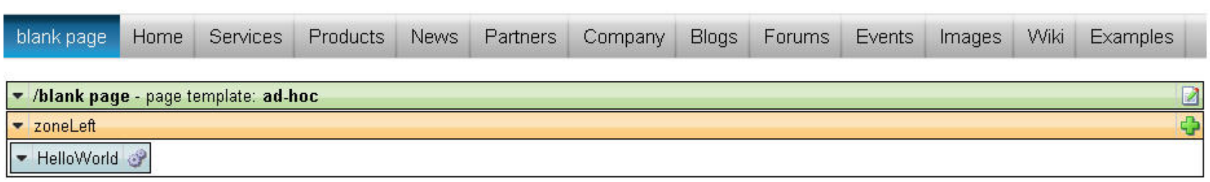
12.Switch to the **Properties** tab and add the following property:
- Attribute name: ButtonText
- Attribute type: Text
- Attribute size: 100
- Field caption: Button text
- Field type: Text Box



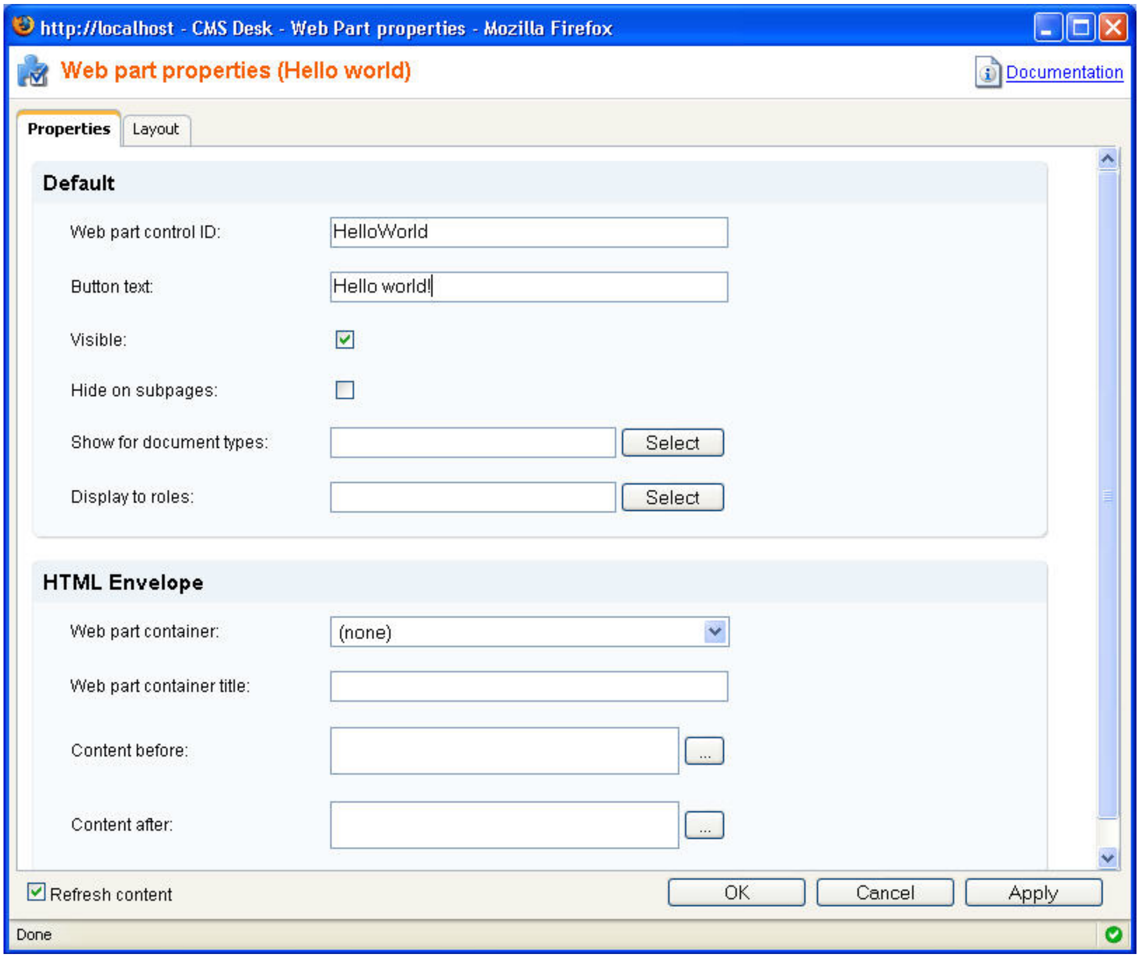

13.Switch to **CMS Desk**.

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

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 window of the **HelloWorld** webpart appears, set the value of the **Button text** field to *Hello world!*




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


You have learned how to create a simple web part.

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

You can also use Kentico CMS Controls on the web part (on 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 in the Design mode - for example:

**[C#]**

```
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 the combination of OnContentLoaded and SetupControl methods. This is the way it is handled in all Kentico CMS web parts. You can see the code of any of the web parts located in *<project folder>/CMSWebParts* and take it as an example.

If you are using classical .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).

## 16.3 Modifying web part behavior

If you need to modify the behavior of standard web parts there are two options:

**1) You only need to set the web part properties dynamically in your code**

You need to create a user control, add the web part into it and write additional code. See chapter Setting web part properties dynamically in your code.

**2) You need to modify the design (layout) of the web part**

You need to use the custom web part layouts described in chapter Customizing web part layout.

**3) You need to modify the code of the web part**

You need to create a copy of the standard web part as described in chapter Modifying the code or design of standard web parts.

## 16.4 Customizing web part layout

Kentico CMS comes with many built-in web parts. The concept of web part layouts allows you to customize the look of the web part by defining a custom HTML layout. So the web part layout is basically a custom skin for the web part.

### Example: Customizing Newsletter subscription dialog

In this example, we will customize the newsletter subscription dialog layout. The standard layout looks like this:


Go to **CMS Desk -> Content** and navigate to the **Home** page (if you're using the sample Corporate Site). Switch to the design mode 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" CodeFile="~/CMSWebParts/Newsletters/
NewsletterSubscriptionWebPart.ascx.cs"
    Inherits="CMSWebParts_Newsletters_NewsletterSubscriptionWebPart" %>

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

<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" />
    <table>
        <tr>
            <td>
                <cc1:LocalizedLabel ID="lblFirstName" runat="server"
AssociatedControlID="txtFirstName" /><br />
                <asp:TextBox ID="txtFirstName" runat="server"
CssClass="SubscriptionTextbox"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>
                <cc1:LocalizedLabel ID="lblLastName" runat="server"
AssociatedControlID="txtLastName" /><br />
                <asp:TextBox ID="txtLastName" runat="server"
CssClass="SubscriptionTextbox"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>
                <cc1:LocalizedLabel ID="lblEmail" runat="server"
AssociatedControlID="txtEmail" /><br />
                <asp:TextBox ID="txtEmail" runat="server"
CssClass="SubscriptionTextbox"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>
                <asp:CheckBoxList runat="server" CssClass="NewsletterList"
ID="chklNewsletters"></asp:CheckBoxList>
            </td>
        </tr>
        <tr>
            <td align="right">
                <cc1:LocalizedButton ID="btnSubmit" runat="server"
OnClick="btnSubmit_Click" CssClass="SubscriptionButton" />
            </td>
        </tr>
    </table>
</asp:Panel>
```

Click **OK**. When you look at the page now, you will see a dialog like this:


**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 Visible="False" attribute to the control, but the control must stay in the layout so that the web part keeps working.

This issue **may also cause problems when upgrading to a new Kentico CMS version** - if some of the built-in web parts uses a new control an you use your web part layout created in the previous version, the web part may stop working. Please be sure to test your web site after an upgrade carefully if you're using web part layouts.

## 16.5 Setting web part properties dynamically in your code

In some cases, you may need to set the value of the web part 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 to this user control. In the user control code, you can implement your custom logic and set the properties appropriately.

The following example shows how you can dynamically set the **WHERE condition (WhereCondition)** property of the **Repeater** web part based on the fact that the current user is or is not authenticated. It uses the standard News document type with custom boolean field **Show to public users (ShowToPublicUsers)**.

1. Open the web project in **Visual Studio**.

2. Create a new folder under the project root called by the code name of your web site, in our case **CorporateSite**. It will ensure that your user controls will be exported with the site when deploying the site to the live server.

3. Create a new user control under the folder **CorporateSite** and call it **NewsRepeater.ascx**.

4. Drag and drop the **CMSWebParts/Viewers/cmsrepeater.ascx** on your user control. You could alternatively use the CMSRepeater server control as well, 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. Add the following code to the code-behind of your user control:

**[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 user is signed in. **Save** all changes.

6. Go to **Site Manager -> Development -> Document types -> News** and add a new field called **ShowToPublicUsers** of type boolean.

7. Go to **CMS Desk -> Content**, choose **Home**, switch to the **Design** mode and a new web part **General/User control** to the **zoneBottom** zone. Set the **User control virtual path** property value to **~/CorporateSite/NewsRepeater.ascx**.

8. Edit some news in the **/News** section of the web site and set the **Show to public users** value to **Yes**.

9. **Sign out** and see the home page. You should see only news items that you marked as **Show to public users**.

You have learned how to dynamically set the web part properties based on your custom logic.

## 16.6 Modifying the code of standard web parts

This chapter explains how you can create a copy of a standard web part and modify its code.

The following example shows how you can send 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 the base and adds a custom handler on the OnAfterSave event.

1. We will create a copy of the BizForm definition in Kentico CMS. Go to **Site Manager -> Development -> Web parts -> BizForms -> BizForm**. Click **Clone web part** and enter the following values:

   - **Web part display name:** BizForm with custom e-mail
   - **Web part code name:** BizFormWithEmail
   - **Web part category:** BizForms
   - **Web part 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. Switch to the **Design** mode and drag and drop a **Label** control on the page. Set its ID to **lblConfirmationMessage**.

3. Click the BizForm control and choose Events in the Properties window. Double-click the OnAfterSave event and add the following method inside the generated method:

**[C#]**

```
protected void BizFormNew_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.BizFormNew.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 `BizFormNew.BasicForm.DataRow` property.

Use valid e-mail addresses and **save** all changes.

4. Go to **CMS Desk -> Content**, choose the **Home** page, switch to the **Design** mode and add the **BizForm with custom e-mail** web part to the **zoneBottom** zone. Set the **Form name** property to the **Contact us** form (if you're using the sample Corporate Site).

5. 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 you can create a modification of a standard web part.

## 16.7 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 it has different default values. It means you can create 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 be later modified for any other value, but the inherited (specialized) web part allows you to do things faster.

### How to create an inherited web part

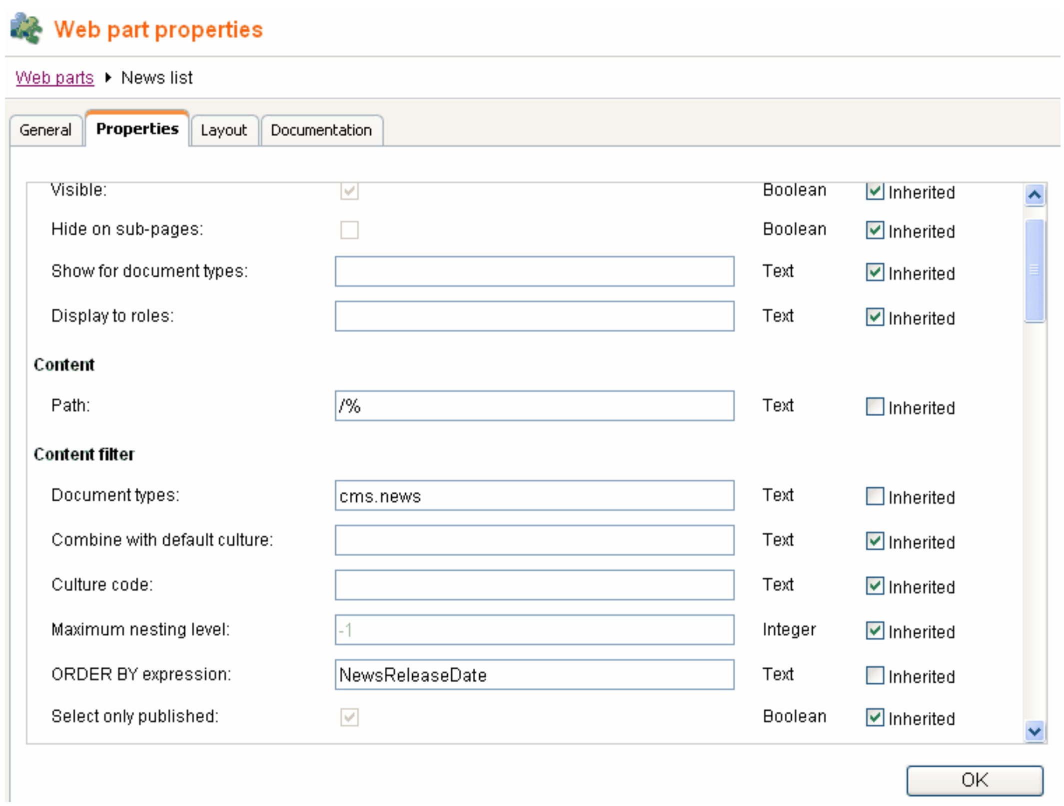
1. Go to **Site Manager -> Development -> Web parts**, click **Listings and viewers** and click **New web part**.

2. Click **Inherit from an existing web part** and enter the following values:
**- Web part display name:** News list
**- Web part code name:** NewsList

**- Inherit from:** Listings and viewers/Repeater
Click **OK**.

3. Click 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

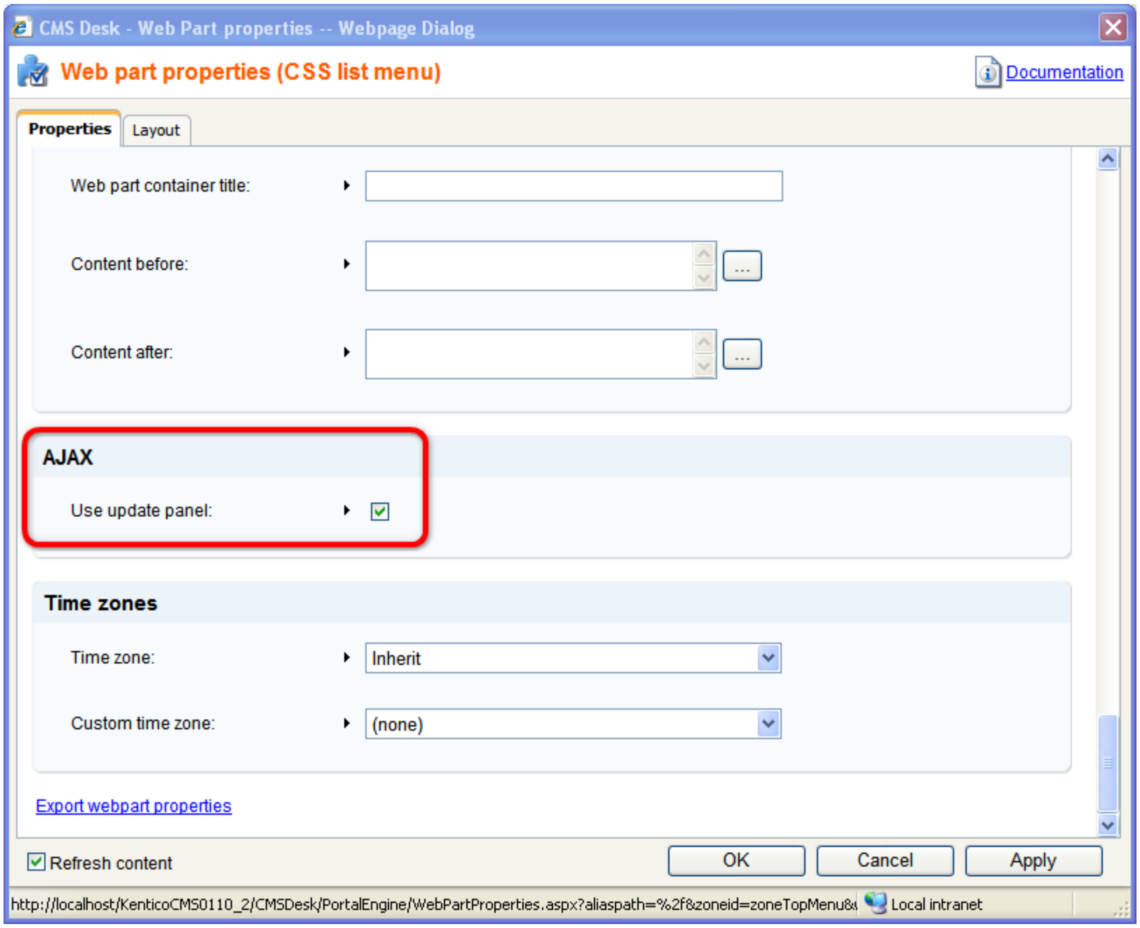
Click **OK**.




4. Go to **CMS Desk**, choose the **Home** page in the content tree and switch to the **Design** mode. Add the News list web part to the page. It will display all site news without any additional configuration now.

## 16.8 AJAX support

Your web parts can use the UpdatePanel, which wraps the web part into the UpdatePanel. This can be easily done by enabling the **Use update panel** web part property of your web part.




No further modifications are necessary, but the following rules should be sticked to:

1) ID of a control must be defined when loading the control dynamically:

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, "");
```

# Part XVI

# 17 Web part containers

## 17.1 Containers overview

Containers are used as "boxes" for web parts. They consist of text that is displayed before the web part and after the web part which means they are used as an envelope for web part content. They have two advantages over using the ContentBefore and ContentAfter properties:

1. They are re-usable on many web parts.
2. They can contain title and dynamically inserted values of the web part properties.

The containers can be managed in **Site Manager -> Development -> Web part containers**.

| | |
|---|---|
| Container display name | The name of the container displayed to the users. |
| Container code name | The name of the container used in the code. |
| Text before web part | Text displayed before web part - the beginning of the envelope. |
| Text after web part | Text displayed after web part - the closing of the envelope. |

The **Text before/after web part** value 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%}`**

Here's an example of web part **without container**:

Your first news (8/3/2006)

*Summary comes here.*

Your second news (8/4/2006)

*Summary comes here.*

Here's an example of the same web part **with container**:


## 17.2 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:
   - Container display name: Blue
   - Container code name: Blue
   - 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. Go to **CMS Desk** and display the design mode of some page. Configure some web part and set its properties like these:
   - Use container: Blue
   - Container title: My web part with container

6. View the page in **Live site** mode. The web part will be surrounded with border and it will have the blue title.

# Part XVIII

# 18 Developing custom modules

## 18.1 Custom modules

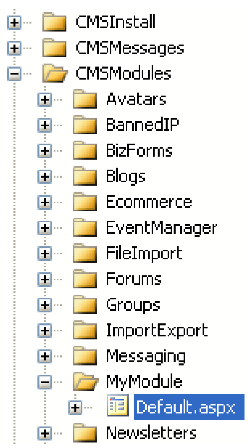
This chapter describes how you can create a custom module integrated into the Kentico CMS administration interface, such as event booking system, CRM system or any other module.

In the following example, you will learn how to create a simple module with a single button that displays the current date and time. You will also learn how to control access to this module.

1. Open Kentico CMS web project in Visual Studio 2005. You can do that either using the WebProject.sln file or using File -> Open -> Web site menu in Visual Studio 2005.

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 module and change the following line:

**[C#]**

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

to:

**[C#]**

```
public partial class CMSModules_MyModule_Default : CMSDeskPage
```

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:

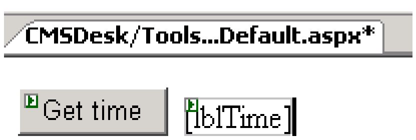
**[C#]**

```
if (!CMSContext.CurrentUser.IsAuthorizedPerResource("myprojects.mymodule", "read"))
{
    RedirectToAccessDenied("myprojects.mymodule", "Read");
}
```

It checks if the current user has 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 Click event handler:

**[C#]**

```
if (CMSContext.CurrentUser.IsAuthorizedPerResource("myprojects.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 module my projects.module and if so, it displays the current date and time.

10.Run the project and sign in to **Site Manager**.

11.Go to **Development -> Modules** and click **New module**. Enter the following values:
- Module display name: My module
- Module code name: myprojects.mymodule
- Show in tools menu: yes
- Resource URL: ~/CMSDesk/Tools/mymodule/default.aspx

Click **OK**.

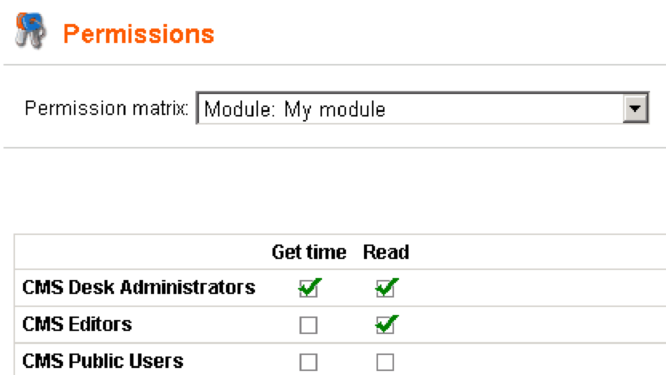
12. Go to the **Permission names** tab and click **New permission**. Enter the following values:
- Permission display name: Read
- Permission code name: read

Click **OK**.

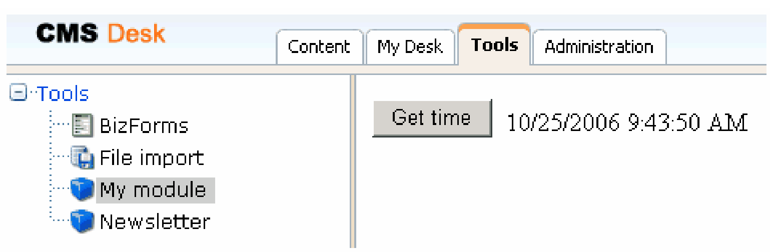
13.Add another permission:
- Permission display name: Get time
- Permission code name: gettime

14.Go to the **Sites** tab and enable the new module to appropriate sites.

15.Go to **CMS Desk -> Administration -> Permissions** and choose the Module: My module value in the drop-down list. Grant **CMS Editors** with **Read permission** and CMS **Desk Administrators** with **Get time** and **Read permissions**:




16.Go to **CMS Desk -> Tools**. You can see the My module icon in the list now. Click it and you will see your form. When you click the button, you can see the current date and time on the right:




You can also try to sign in under a different account with different permissions and test the access control.

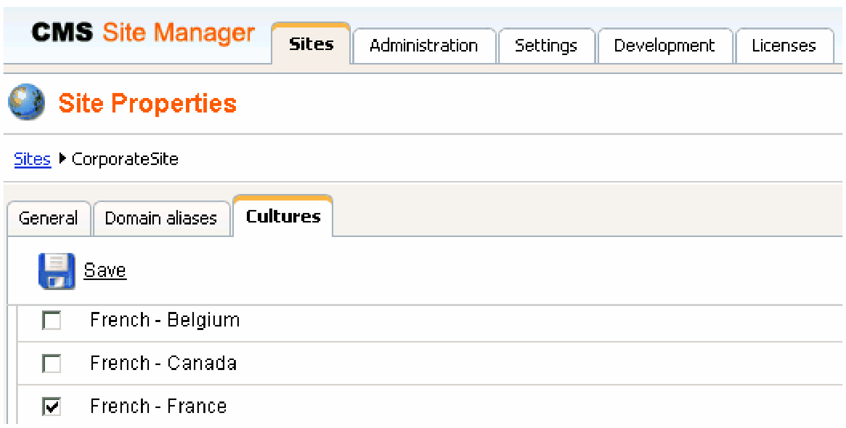
# Part XIX

# 19 Multilingual and international support

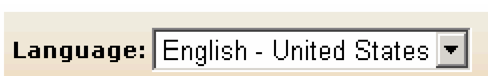
## 19.1 Configuring multiple languages

This chapter explains how to configure web site for displaying content in multiple languages. When you create a new web site, it uses the culture you specified during the New site wizard. The following guide shows you how to add extra language versions to the standard Corporate Site:

1. Go to **Site Manager -> Sites** and click **Edit site** button next to the Corporate Site. Go to the **Cultures** tab. Check the box next to the following cultures: French - France and German - Germany. Click **Save**.

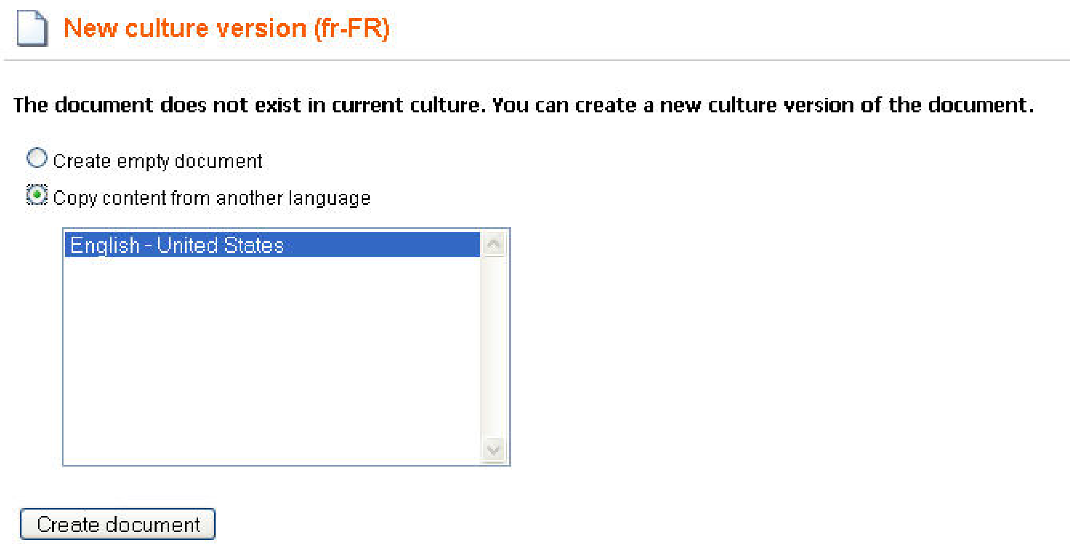



2. Switch to the CMS Desk. You can see there's a new drop-down list with language selection in the main toolbar:

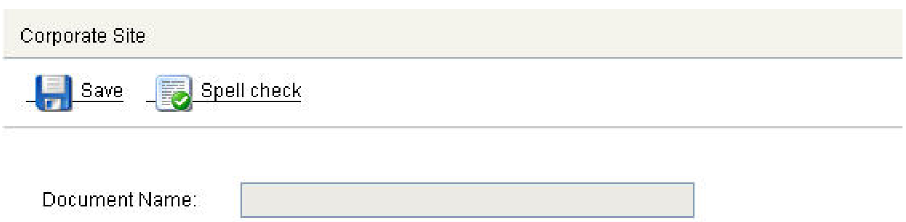



3. Click root document and choose **French** culture. Since the French version of the document doesn't exist yet, you're offered with two options:
- **create empty document** - this option creates a new document of the same type without any content.
- **copy content from another language** - this option 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**:

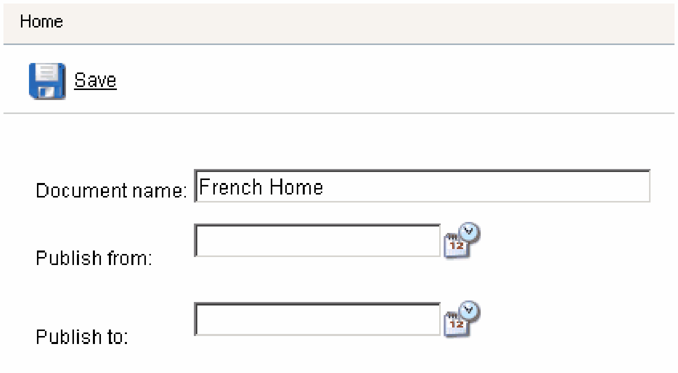



4. You are redirected to the editing form of the new version:



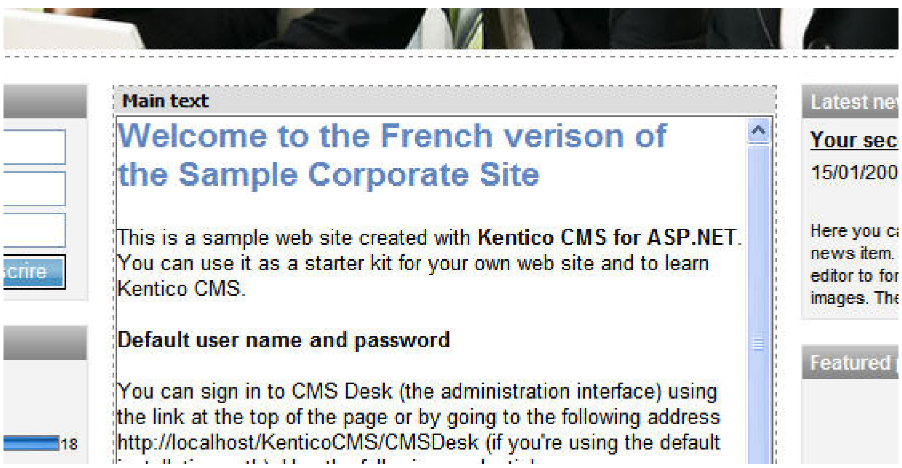

The root cannot have any name, so the box is now disabled. Click **Save**. The new document culture version is created.

5. Now repeat the same for the **Home** page. On the editing form, change the document name to: French Home.



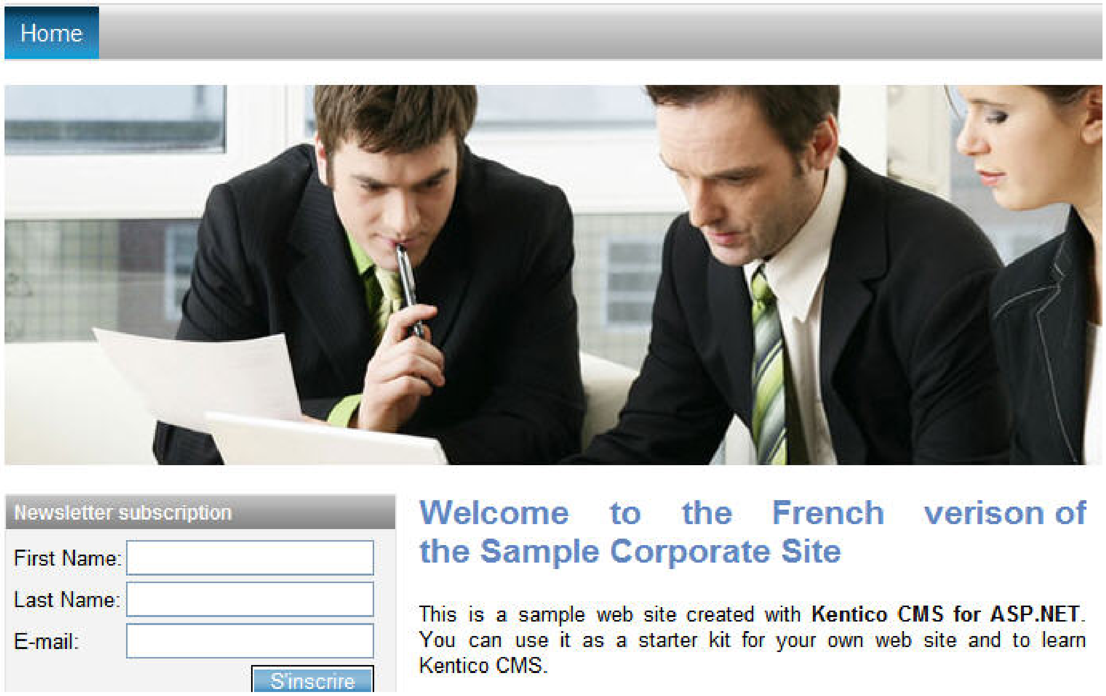

Click **Save**.

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 -> Web site**. 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 web site 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 web site 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.

## 19.2 Default language selection

When the user first comes to the web site, her 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 web site.

### 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:

- mydomain.com (main domain)
- mydomain.de (domain alias)
- mydomain.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. mydomain.fr domain, the content is automatically displayed in French.

## 19.3 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 has always this URL: **`/Home.aspx`**

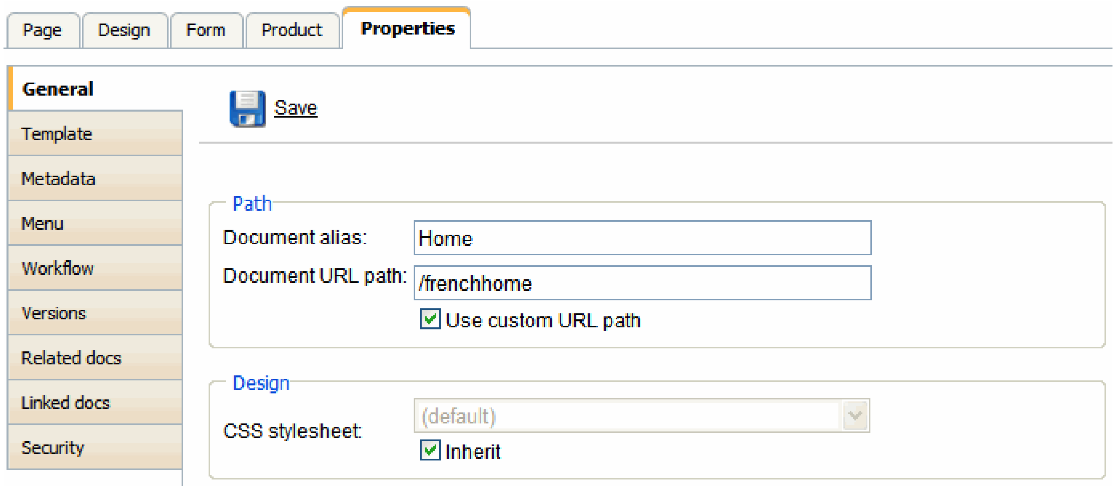
If you want to see the same page in French, you need to go to: **`/Home.aspx?lang=fr-fr`**

Once the language is changed, the selected language is stored in the visitor browser's cookie and when the user comes back to URL /Home.aspx (without any parameter), she is displayed with the French version.

### 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 web site culture is automatically switched to French and you're displayed with French version.

## 19.4 International and RTL support

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 web site 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 web site.

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**

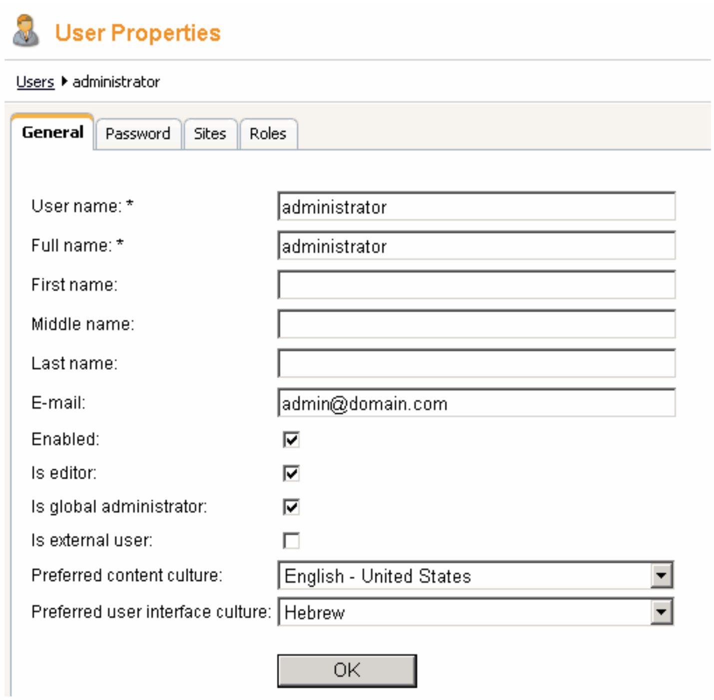
When looking for a localized strings, the system uses the following priority:

1. cms.resx
2. custom.resx
3. database (Site Manager -> Development -> UI Cultures)

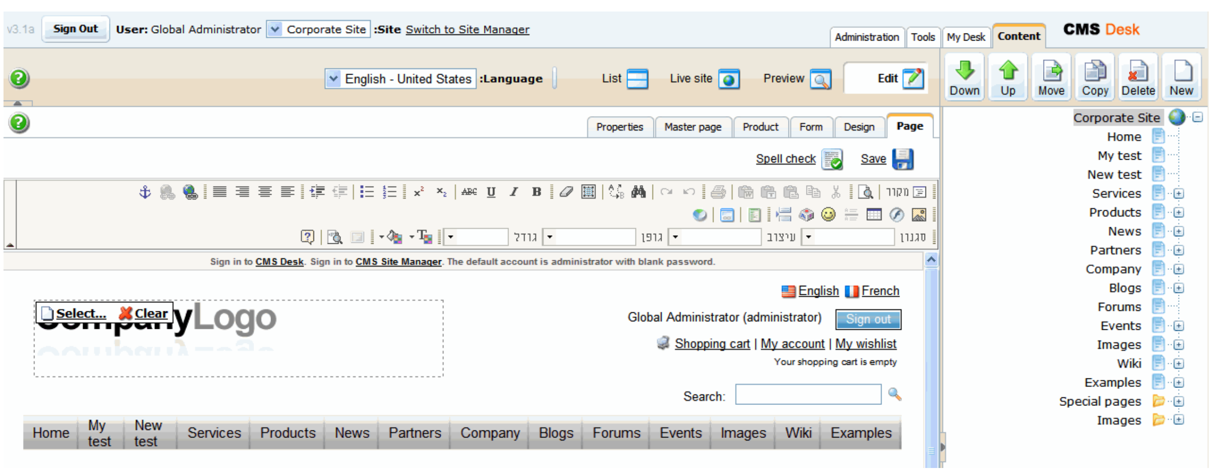
It means if there are duplicate strings with same name in all three files, the system will use the one stored in the filesystem (cms.resx).

## 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):



## Using the Right-to-left languages

If the web site is displayed in language that uses right-to-left direction, 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.

**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.

**UI culture vs. content culture**

The user interface direction is driven by the preferred UI culture of the current user while the content direction is driven by the preferred (content) culture of the current user.

## 19.5 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:**<br><br>`{$key$}` | 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:`**<br><br>`{$=default_string\|culture_code=translation\|culture_code=translation\|etc.$}` | Displays the strings defined in the expression.<br><br>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

# Part XX

# 20 Caching and performance

## 20.1 Performance overview

The performance of your web site depends on many aspects:

1. Hardware on which your web site and database server are running.
2. Available system performance when you're sharing the system resource with other applications (typically in shared hosting environment).
3. The size of your web site.
4. The complexity of the web site (number of nesting levels, number of web parts on the page, etc.)
5. Custom code you added to the web site.
6. Use of caching.
7. Other special circumstances, such as network connectivity between web server and SQL server, etc.

### Troubleshooting Kentico CMS 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 web site.**
Go to CMS Site Manager -> Settings, choose the appropriate web site in the Sites drop-downlist and choose the Web Site 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 web site may be slowed down by output filters. Go to Site Manager -> Settings, choose the appropriate web site in the Sites drop-down list and choose the URLs 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 web site:

- 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 web site 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 web site, 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/web sites 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 web site 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 web site and we will try to analyze it.

## 20.2 Caching options

### Global caching settings

Go to **CMS Site Manager -> Settings -> Web Site**, choose your web site 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.

## Web part/control-level caching settings

Some web parts/controls used for displaying content have two properties related to caching:

- **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 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.

## 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.

## Caching and personalization

If your web site contains sections for site members, the caching will be personalized, which means each signed in user will have his/her 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.

## File management and performance

The files can be stored in 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.

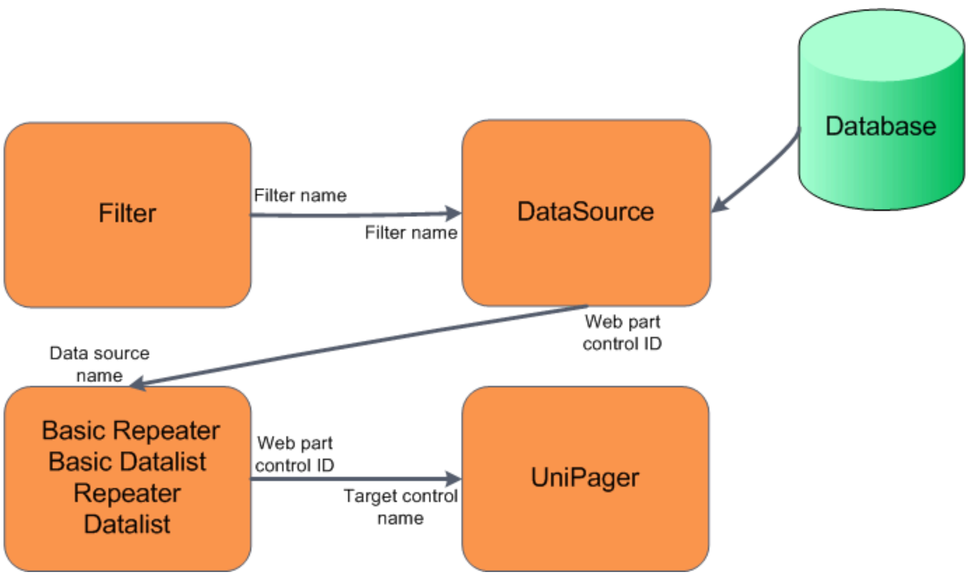
# Part XXI

# 21 Data sources

## 21.1 Using DataSource web parts

DataSource web parts are designed for retrieving data from the database and sending it to other web parts that will 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 in comparison with using the original Repeater, Datalist, etc.

The following diagram shows how a DataSource 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.




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 DataSource concept works in practice.

1) Sign in to CMS Desk as the administrator (login *administrator* with blank password).

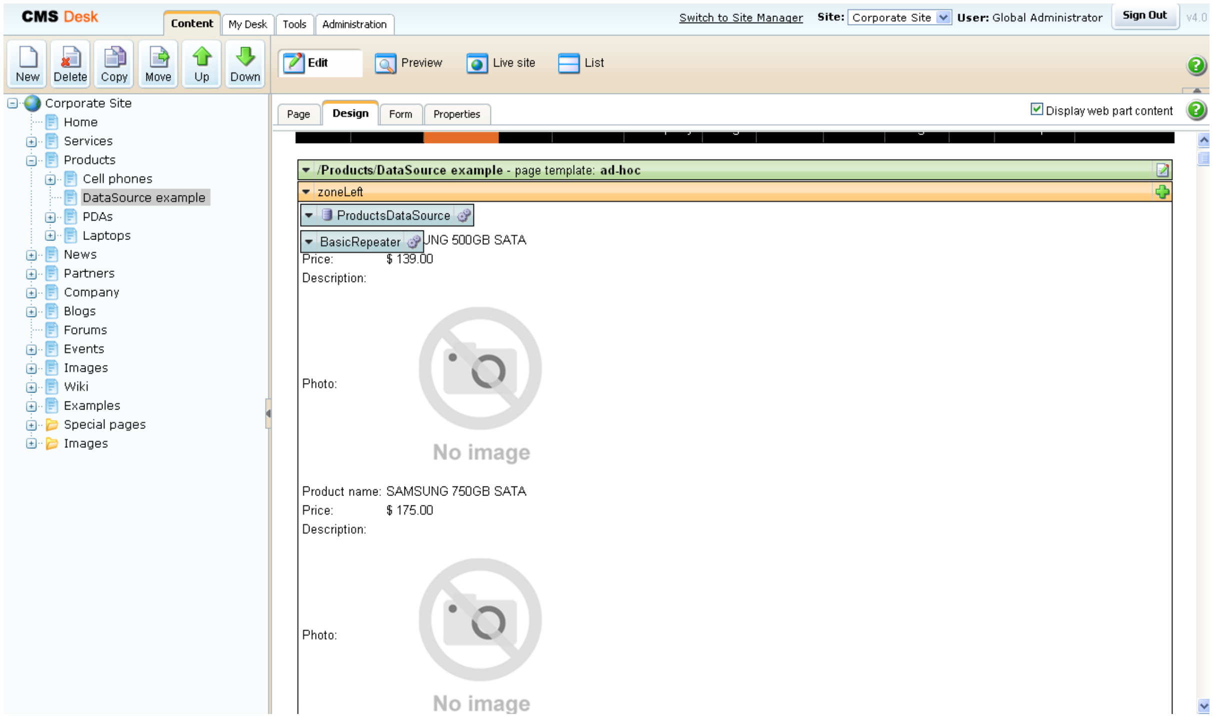
2) Create a new page under the **Products** section. Name it **DataSource example** and choose to **Create a blank page** using the **Simple** layout. Click **Save**.

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**.

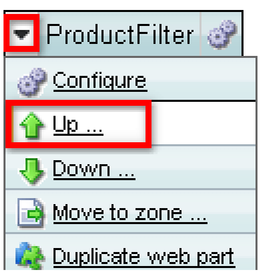
4) Add the **Listings & Viewers -> Basic Repeater** web part to the same web part zone. Set the following properties:

- **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 that you will 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:

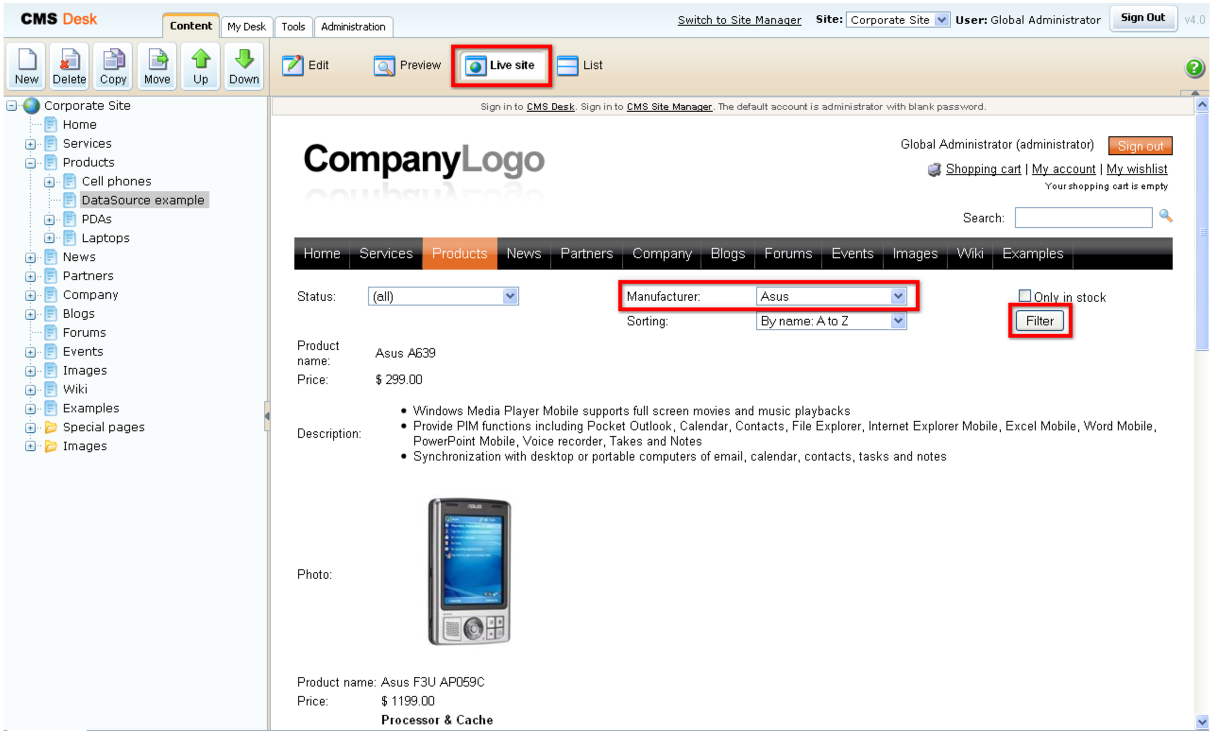
- **Filter name**: ProductFilter

Also open the ProductsDataSource's web part properties and enter the same filter name into its **Filter name** filed:

- **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) 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. 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 -> UniPager** 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:

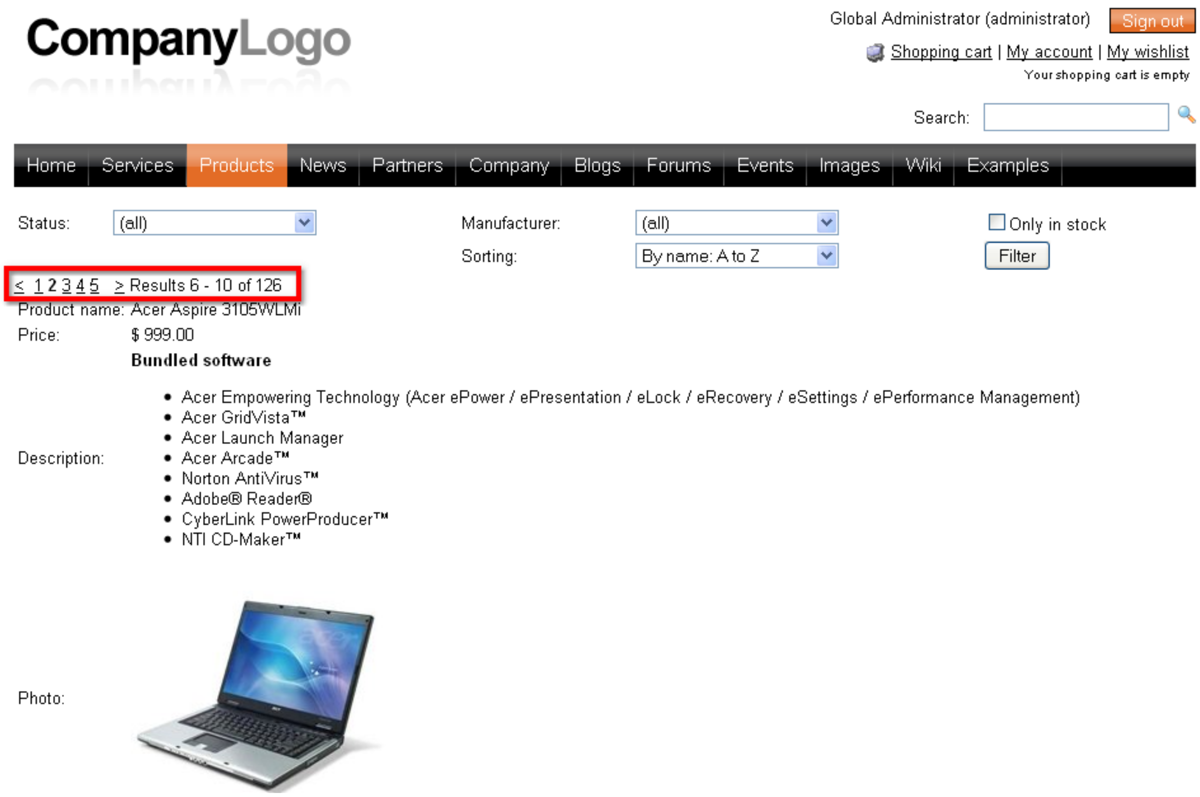
- **Target control name**: BasicRepeater
- **Page size**: 5
- **Group size**: 5

and the following transformations:

- **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, size of the page should be reduced to 5 products and you should be able to switch between pages using the pager.



# Part XXI

# 22 Membership and User registration

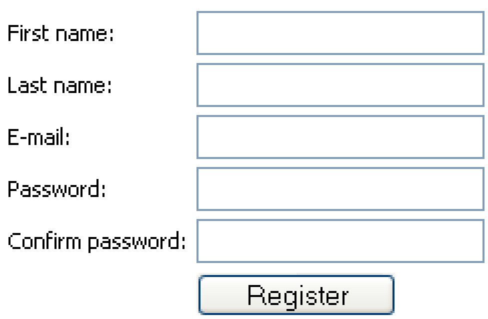
## 22.1 Available registration web parts

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.

## 22.2 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 to any page of your web site 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.

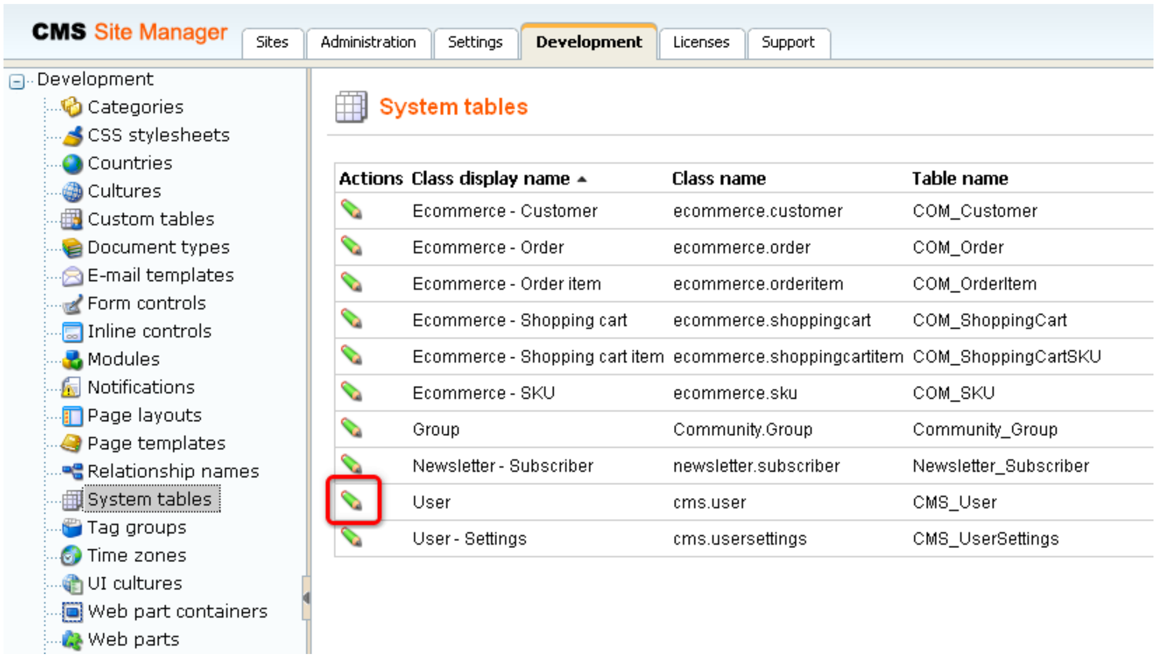



## 22.3 Creating a custom registration form

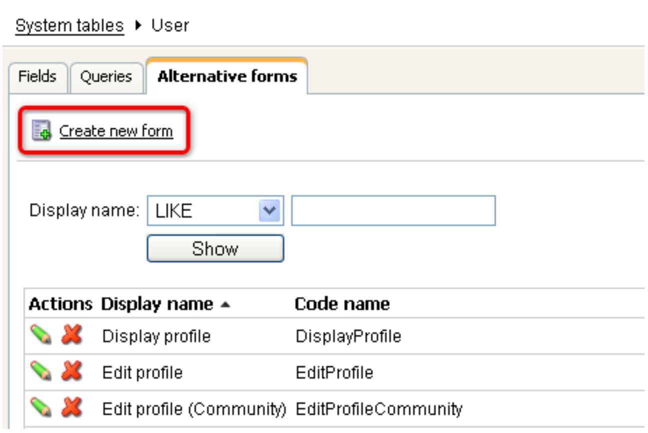
The **Custom registration form** web part can be used in situations when you want to use a different registration form than the default one provided by the **Registration form** web part. This is typically when you want users to provide different details or when you want to customize the form's layout.

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

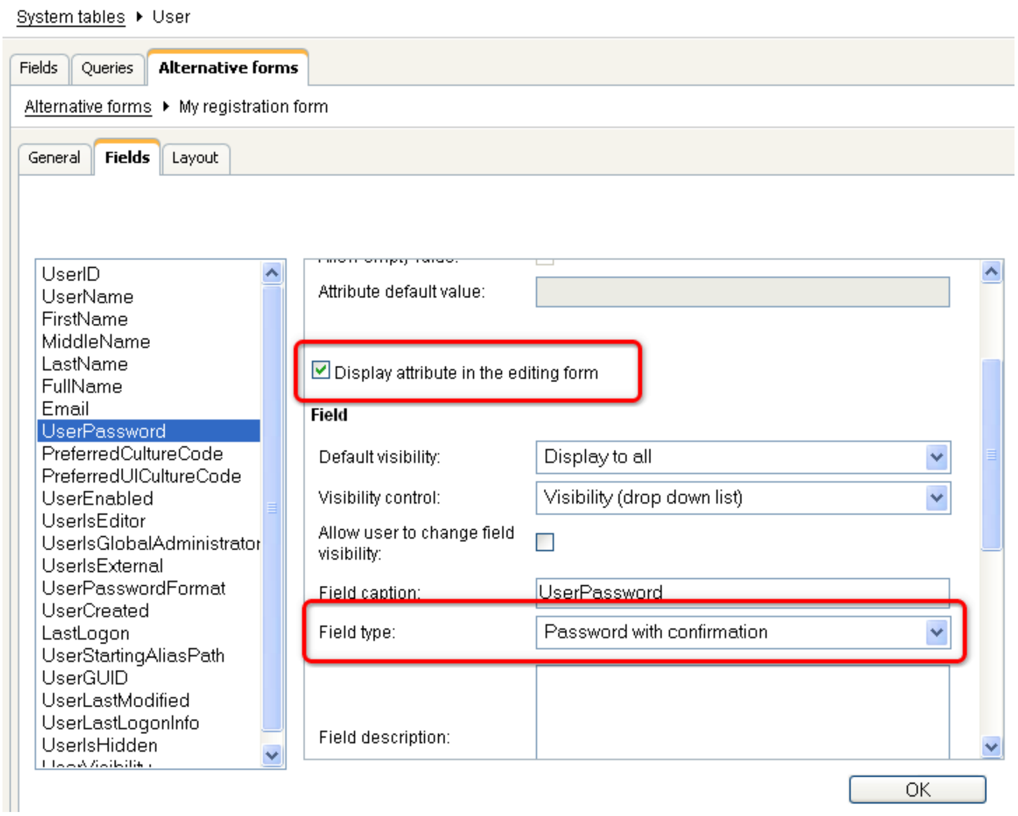
and click **OK**.


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:

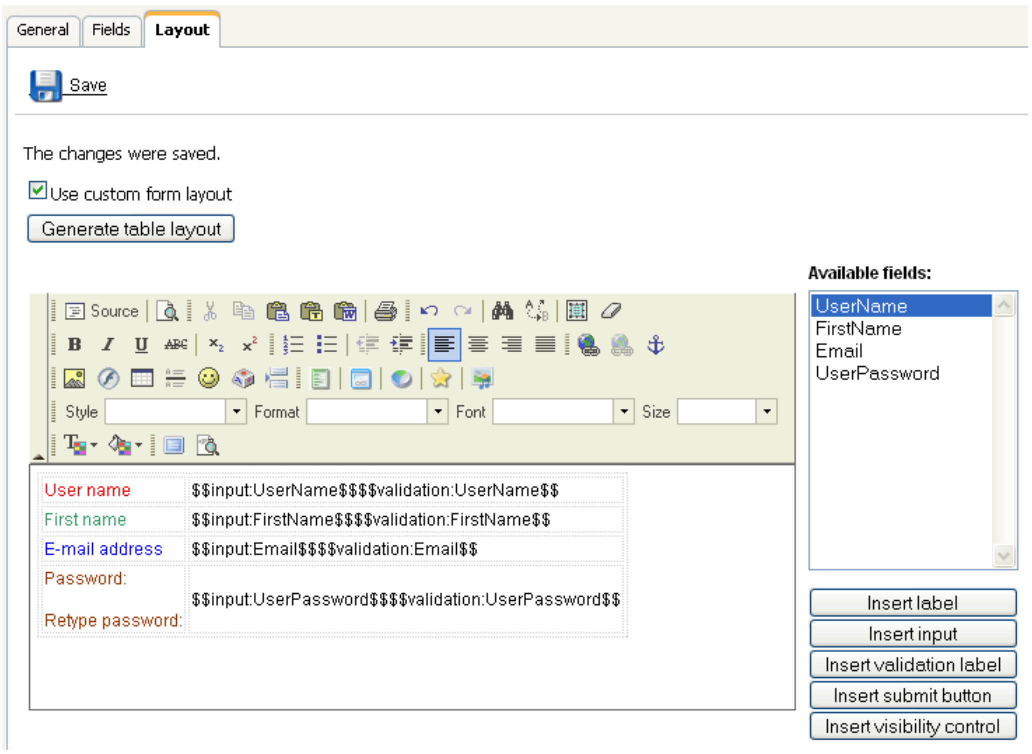
- **UserName**
- **FirstName**
- **Email**
- **UserPassword**

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, change also 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.

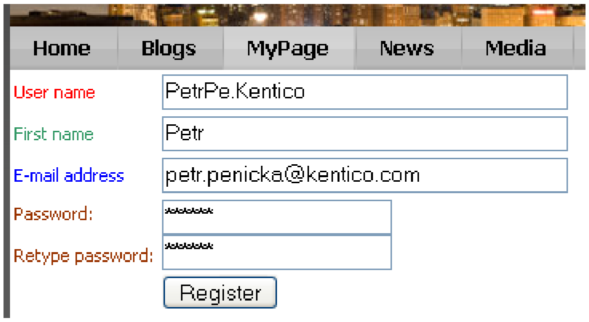
To do it, check the **Use custom layout** check box and click the **Generate table layout** button. A default layout will appear in the editor. Try playing around a bit with  the layout or just create something similar to what you see in the screenshot below. Click **Save** when you are finished.




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** mode and add ( ) the **Membership -> Custom registration form** web part to the page. Set the following property of the web part:

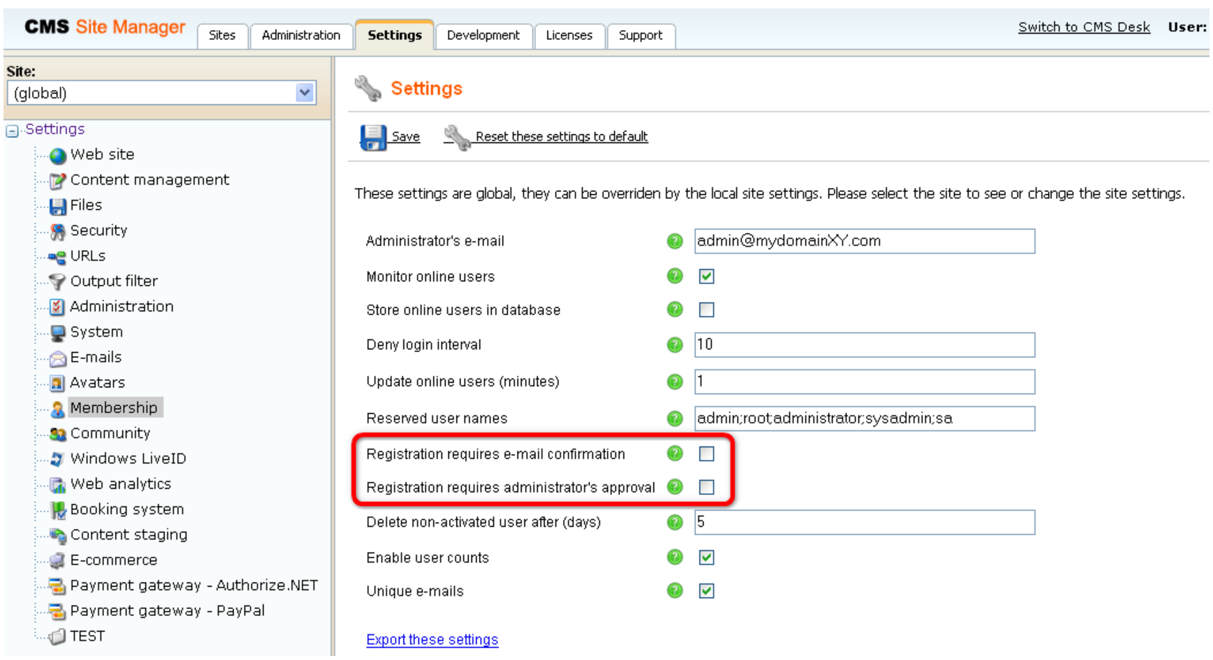
- **Alternative form** - cms.user.MyRegistrationForm

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 web site using the form and verify that the user has been created in **Site Manager -> Administration -> Users**.



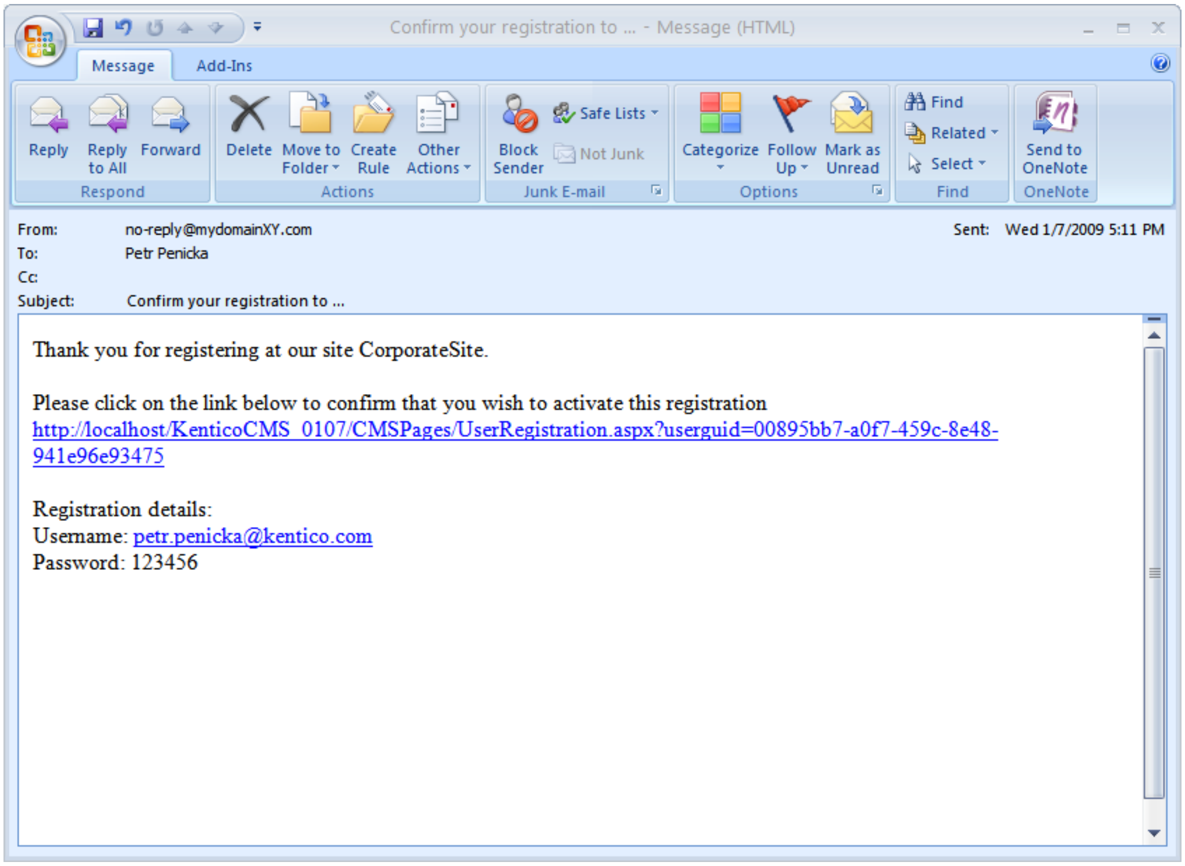

## 22.4 Registration approval and double opt-in

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 user will receive confirmation e-mail to the e-mail address specified on 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** e-mail template.



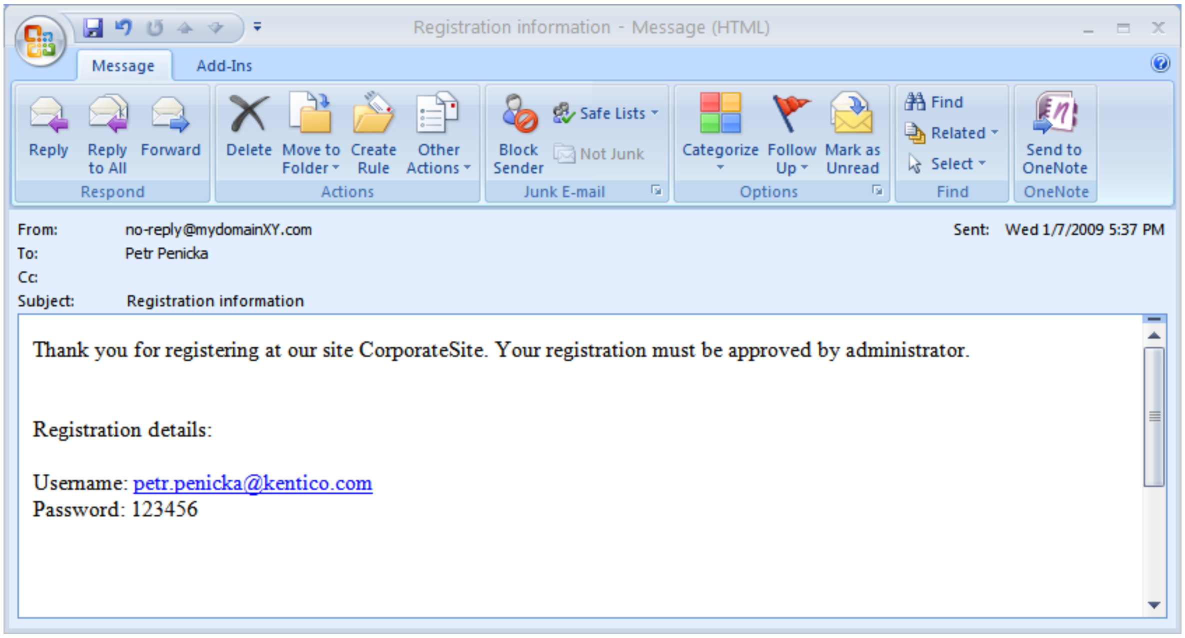

After clicking the link, a special page will be opened in user's web browser, displaying the following message:

Your registration has been approved. Now you are able to login using your username and password. Click here to continue.

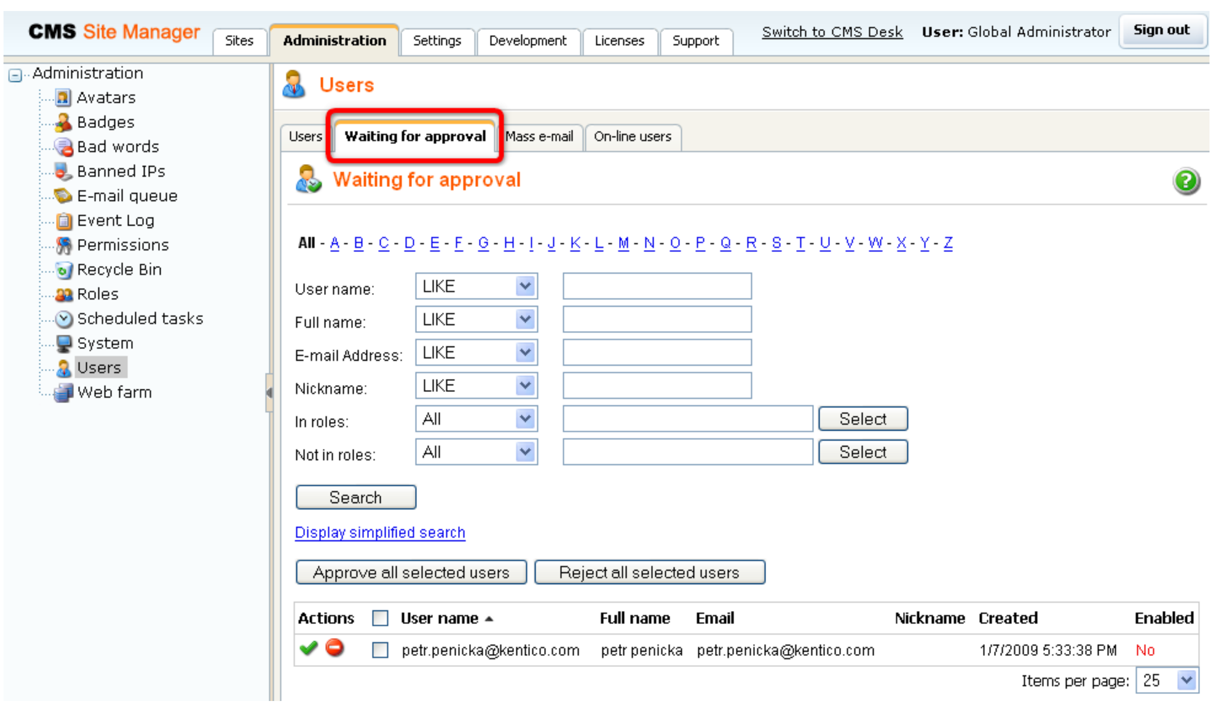
The link at the end of the message will redirect the user to the title page of the web site. The user can then log in using the registration details received in the e-mail.

## Registration requires administrator's approval

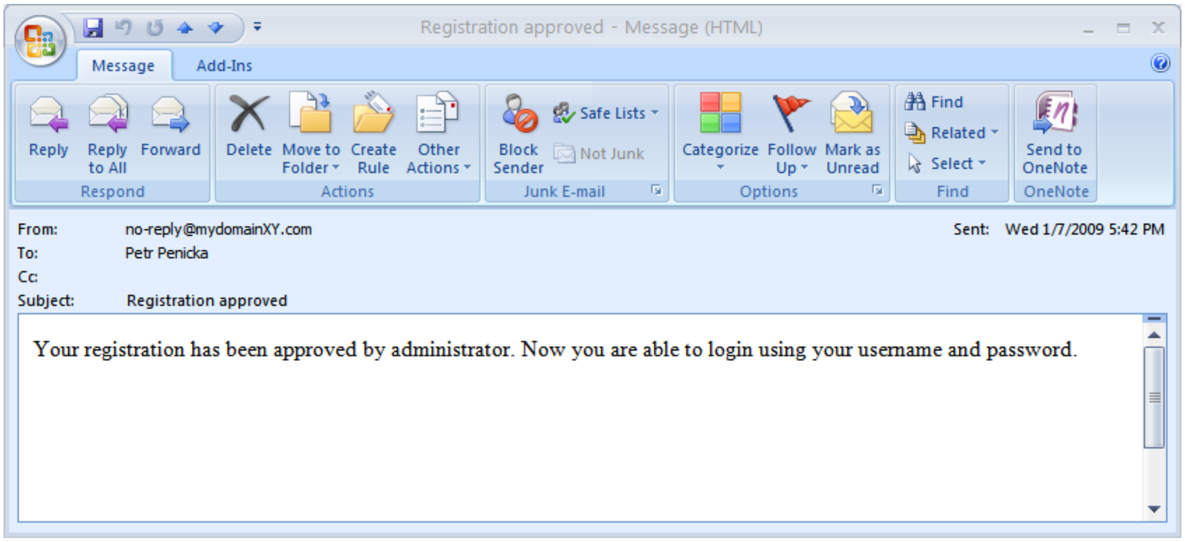
If this option is enabled, users will not be able to sign-in immediately after registration. Their registration will have to be approved by site administrator. At this point, users will receive an e-mail based on the **Membership - Registration waiting for approval** e-mail template. You can see the default version of the e-mail in the screenshot below.




In case that 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** ( ) user's registration.



After the administrator's approval, user receives another e-mail, confirming that her 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, e-mail with the confirmation link will be sent first. After 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 set the **Redirect to URL** property of the registration web part properly. This means that users should not be redirected to any page displaying information about their user account (just as 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 page would display an error message, which might be misleading for the users.

## 22.5 Custom fields visibility

### 22.5.1 How it works

The visibility controls functionality is designed to enable registered users 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**.

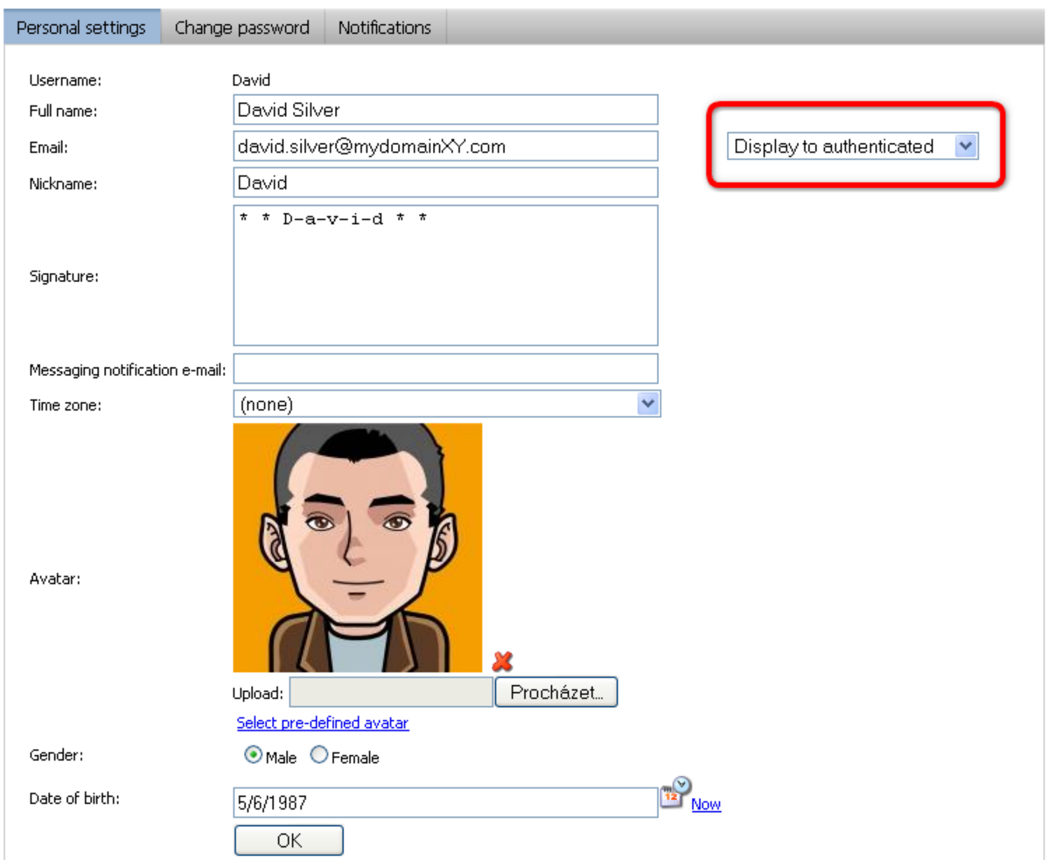
1) Run the **Community site** and sign out of the administration interface. Log on to the site using the **Sign in** form on the right. Enter *David* with blank password and click **Log on**.

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:

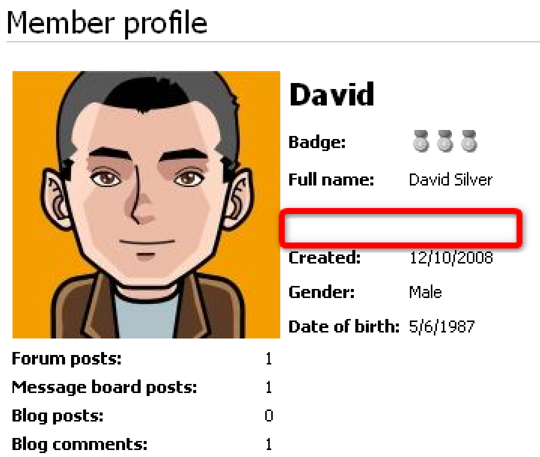
**Display to none** - the field will not be displayed to anyone
**Display to all** - the field will be displayed to everyone
**Display to authenticated** - the field will be displayed only to authenticated users
**Display to friends** - the field will be displayed only to the user's friends

Choose **Display to authenticated** and click **OK**.

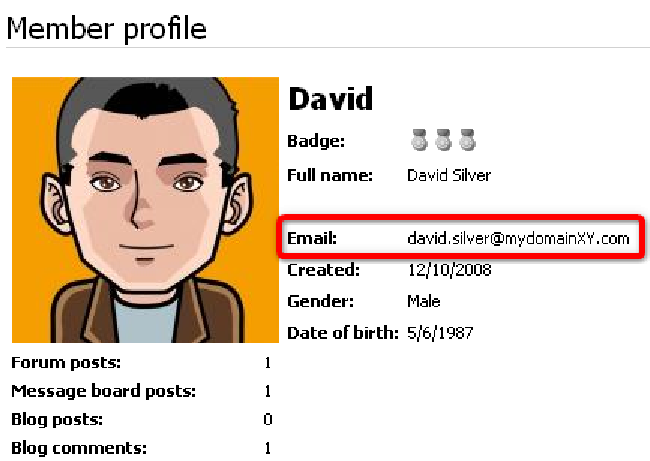



4) You have just configured the user's profile so that only authenticated users can see his e-mail address. Let's verify that it really works. Sign out and visit David's profile as an unauthenticated site

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.



## 22.5.2 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 show in the example in chapter How it works. 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.

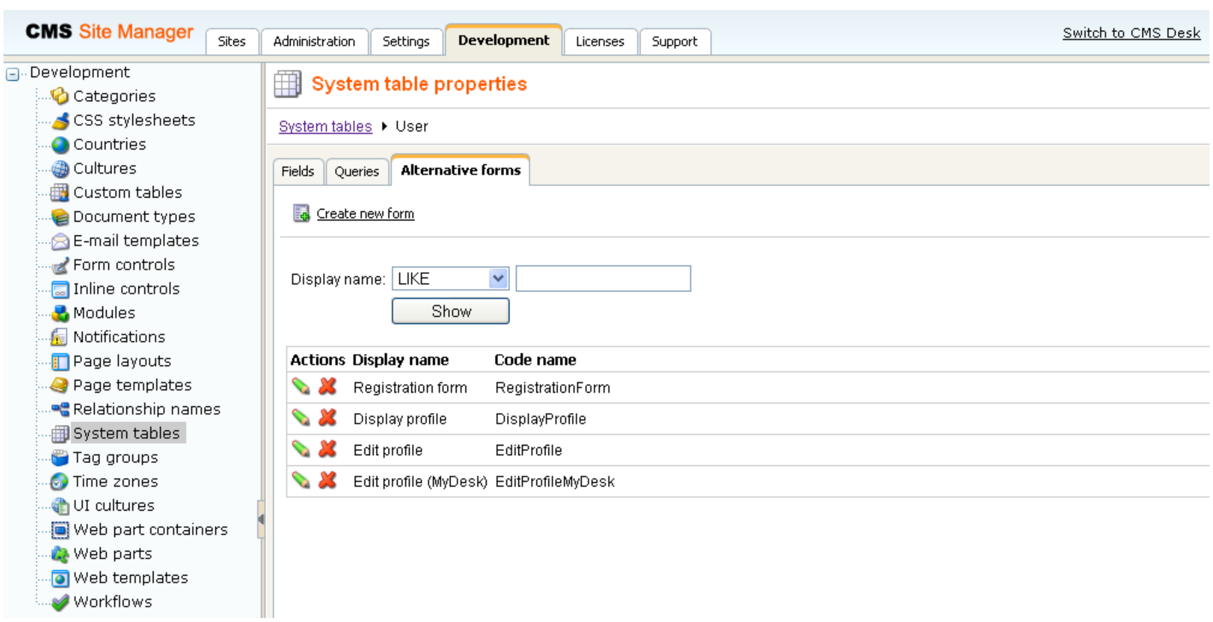
The **CMS_User** system table is the database table where information about registered users is stored. Each of the four alternative forms that you see in the list is used in a specific situation when the system accesses the table:

- **Registration form** - when registering a new user using the **Custom registration form** web part
- **Display profile** - displaying user's public profile using **User public profile** web part
- **Edit profile** - when editing user's profile using the **My account** web part
- **Edit profile (MyDesk)** - when editing 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:

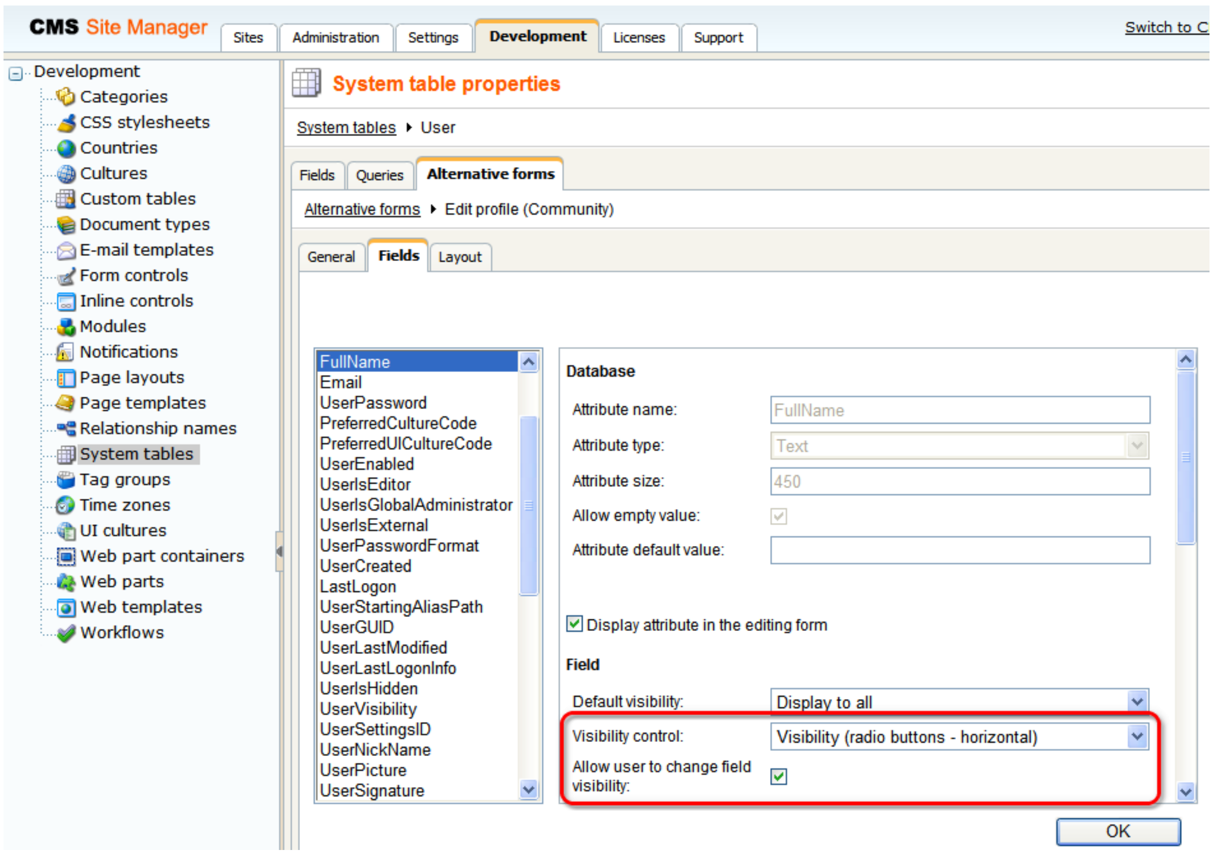
- **Display profile** - displaying group's public profile using **Group public profile** web part

Field visibility 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 (radi buttons - horizontal)** from the **Visibility control** drop-down. Check the **Allow user to change field visibility** check-box and click **OK**.

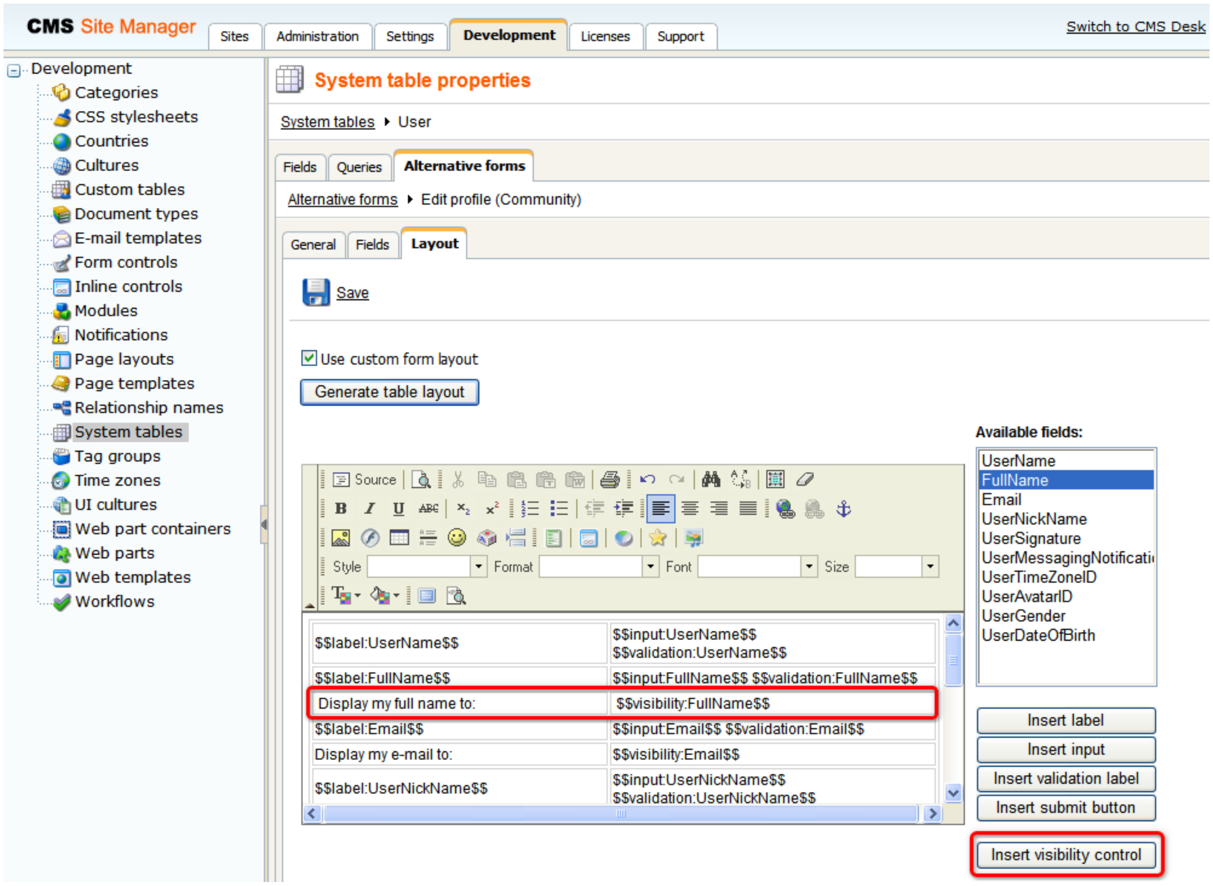


2) Switch to the **Layout** tab. We will need to create another line in the table and add the visibility control into it.

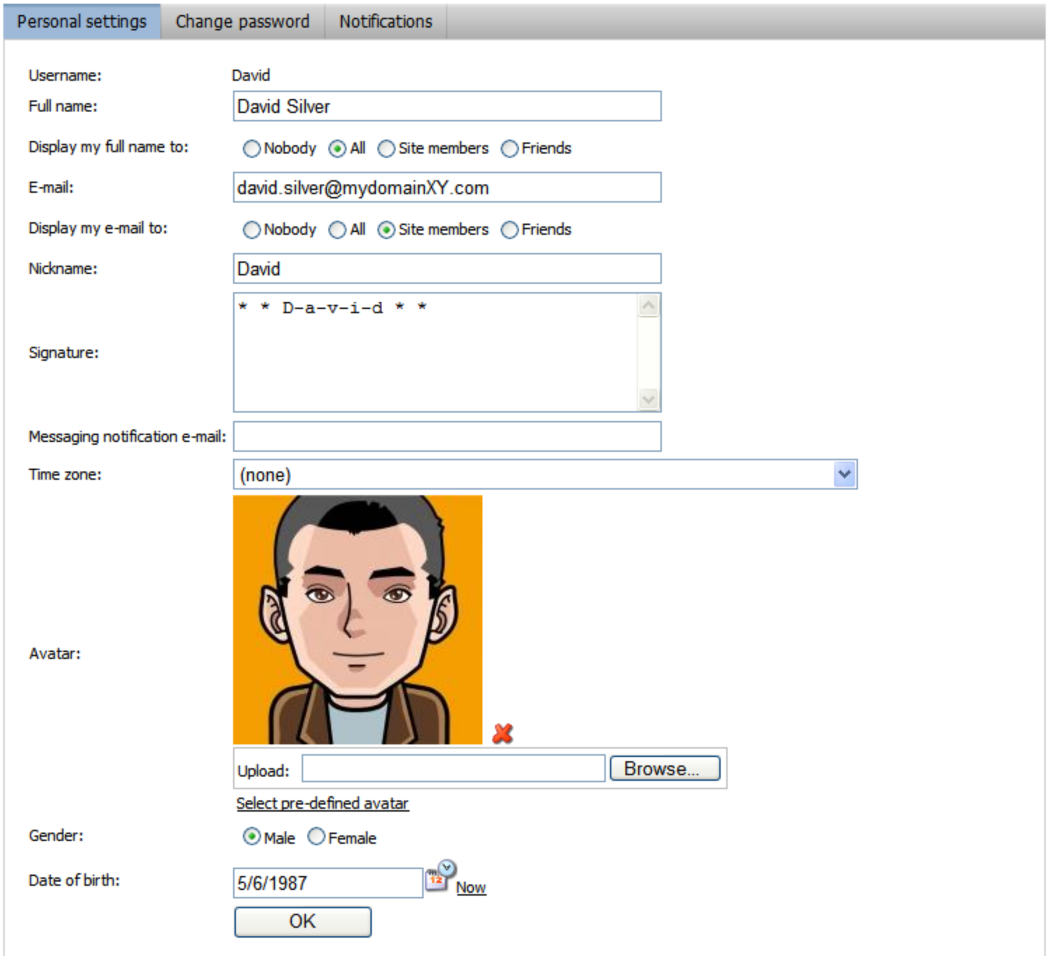
Place the cursor into the FullName line (where the $$label:FullName$$ value is). Right-click and choose **Row -> Insert Row After** from the context menu. Into the first column of the new row, enter *Display my full name to:*.

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**.


CMS Site Manager
Sites
Administration
Settings
Development
Licenses
Support
Switch to CMS Desk
Development
Categories
CSS stylesheets
Countries
Cultures
Custom tables
Document types
E-mail templates
Form controls
Inline controls
Modules
Notifications
Page layouts
Page templates
Relationship names
System tables
Tag groups
Time zones
UI cultures
Web part containers
Web parts
Web templates
Workflows
System table properties
System tables ▸ User
Fields
Queries
Alternative forms
Alternative forms ▸ Edit profile (Community)
General
Fields
Layout
Save
Use custom form layout
Generate table layout
Available fields:
UserName
FullName
Email
UserNickName
UserSignature
UserMessagingNotificati
UserTimeZoneID
UserAvatarID
UserGender
UserDateOfBirth
Source
Style
Format
Font
Size
$$label:UserName$$
$$input:UserName$$ $$validation:UserName$$
$$label:FullName$$
$$input:FullName$$ $$validation:FullName$$
Display my full name to:
$$visibility:FullName$$
$$label:Email$$
$$input:Email$$ $$validation:Email$$
Display my e-mail to:
$$visibility:Email$$
$$label:UserNickName$$
$$input:UserNickName$$ $$validation:UserNickName$$
Insert label
Insert input
Insert validation label
Insert submit button
Insert visibility control

2) Now sign out of the administration interface and sign in to the web site 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.




## 22.5.3 Use in custom form layouts

If you want to define custom form layout and use the visibility drop-down list for some field, you have to do the following two things:

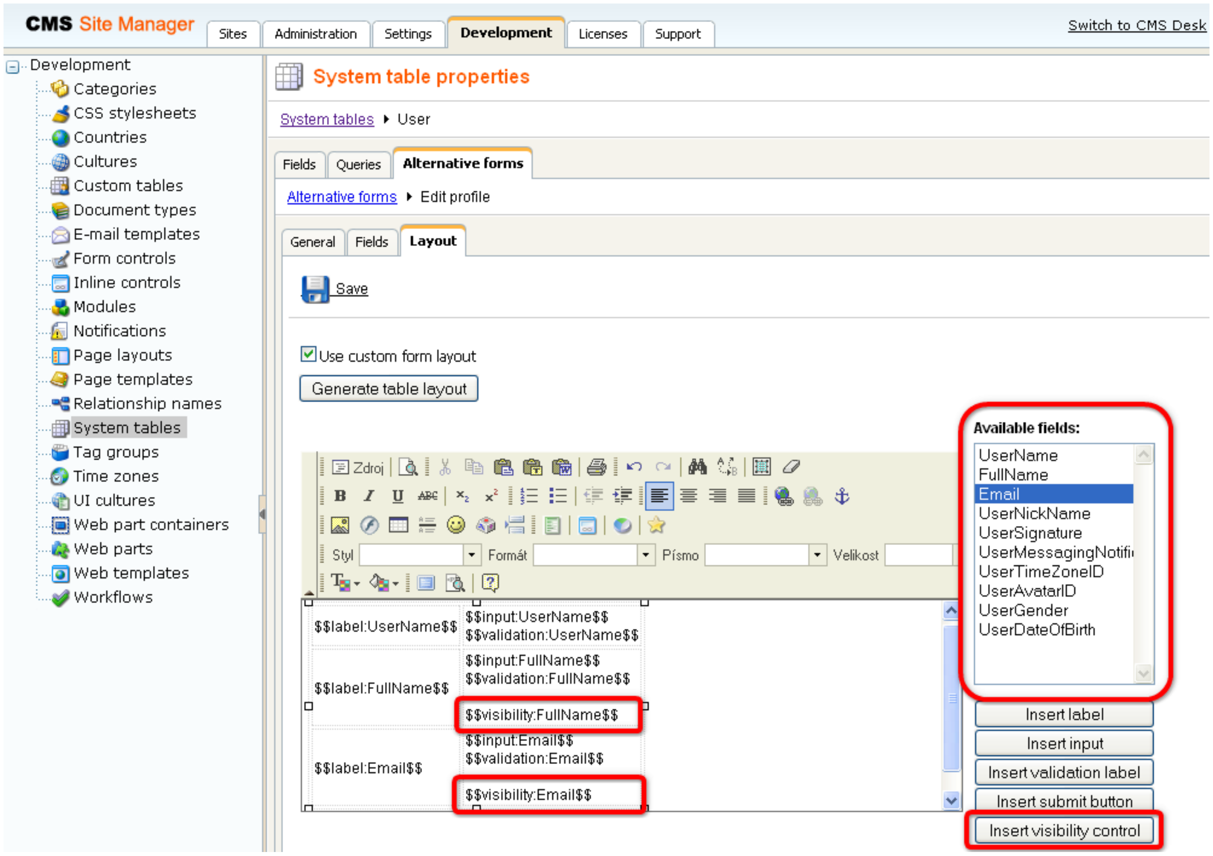
- the **Allow user to change visibility** check-box must enabled for each field that you want to use the drop-down list for. In case it is not enabled, the drop-down list will not be functional.

- 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 **Atlernative 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.




## 22.5.4 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

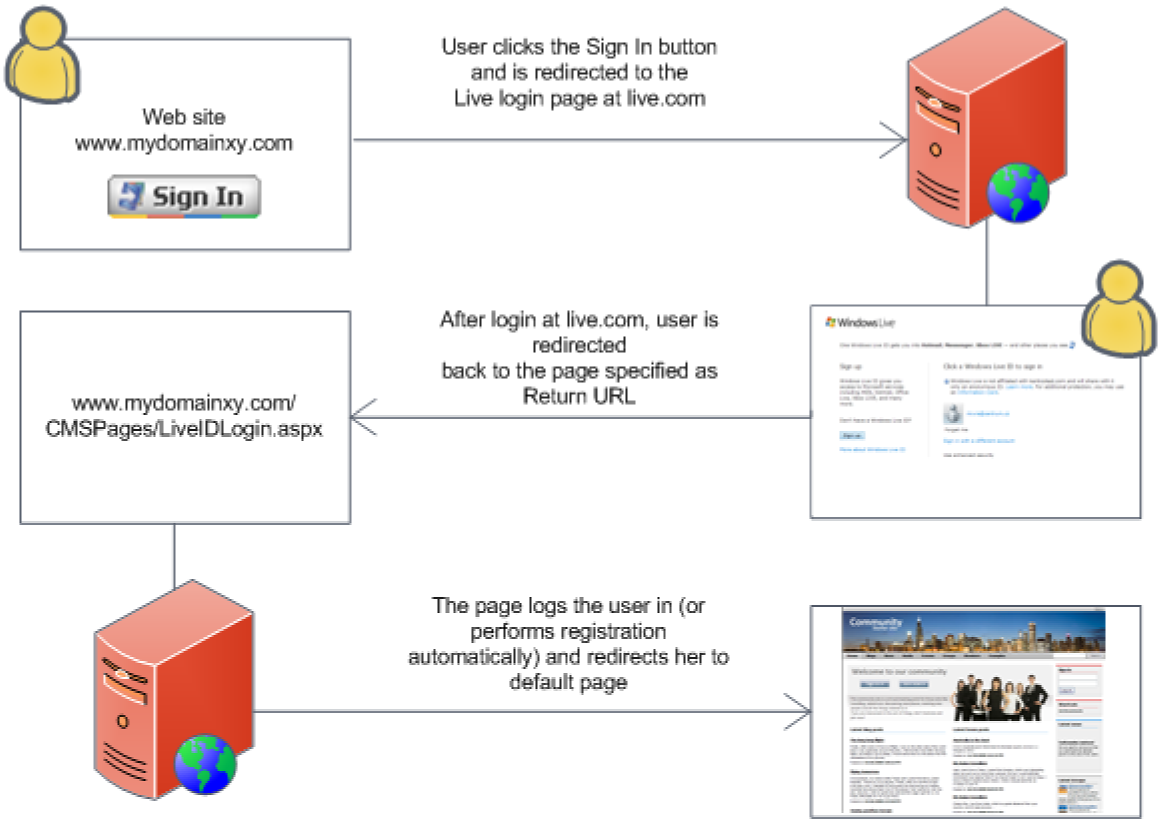
- **Form name** - specify the full name of the desired alternative form (*cms.user.EditProfile* by default)
- **Allow user to edit field visibility** - check to enable the visibility controls

# 22.6 Windows Live ID

## 22.6.1 Overview

LiveID is a single sign-on service provided and maintained by Microsoft. By integrating LiveID into your web site, you can allow site visitors to log in to your web site using their LiveID login and password. Even unregistered users can log in using their LiveID, in which case a new user account will be created automatically based on their LiveID account.

The following diagram shows how the process of LiveID login works.




For this to work on your site, you have to do the following things:

1) Register your web site at https://msm.live.com/app - to learn how to do it click here

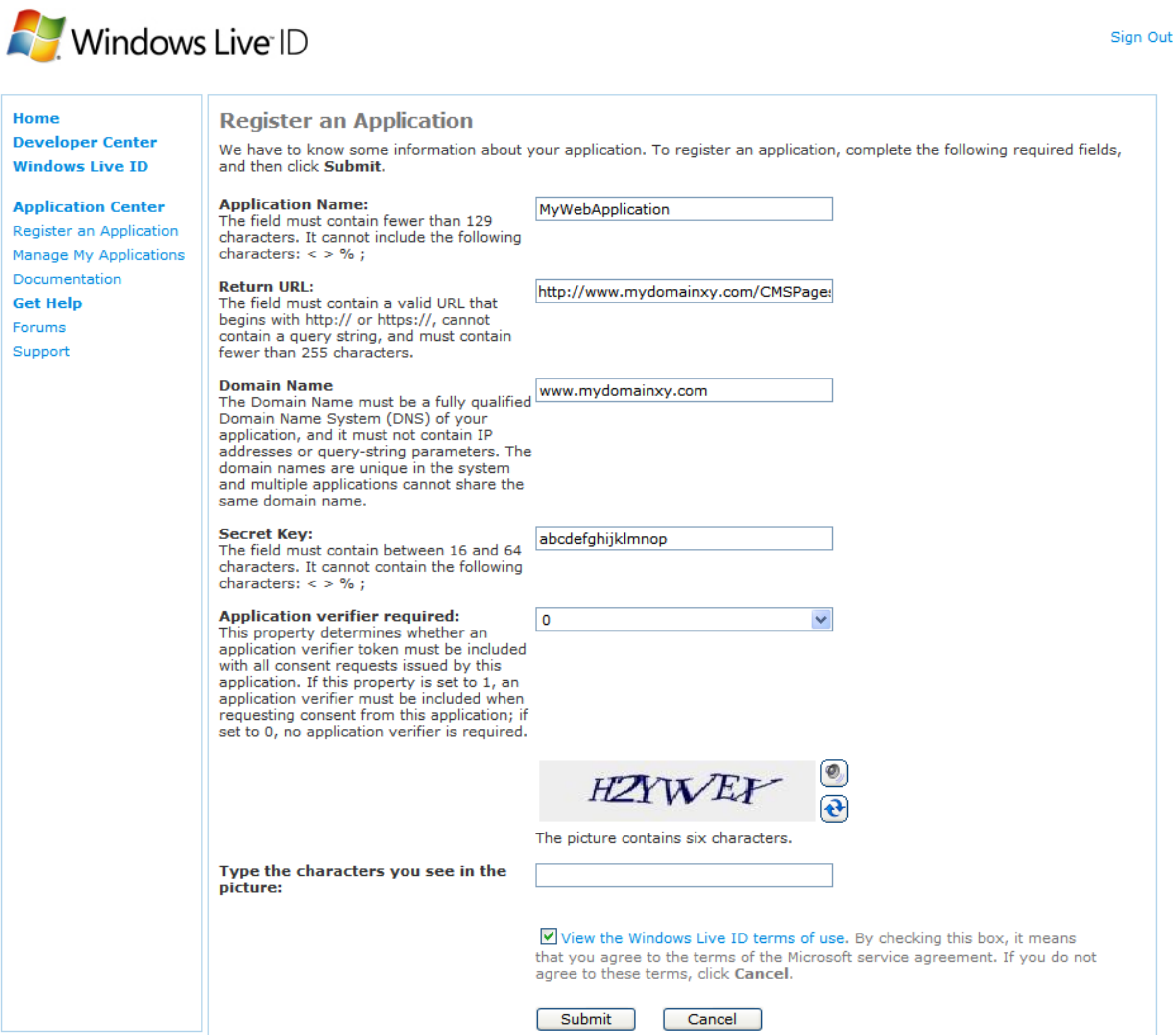
2) Setup kentico CMS LiveID support - to learn how to do it click here

3) Add the LiveID web part to your site and optionally add and configure the Live ID required data web part - info here

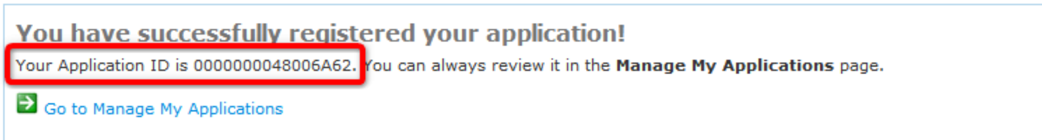
## 22.6.2 Registering your application

To enable the LiveID log on for your site, you must register your web site at the following address: https://msm.live.com/app. Click the **Register an application** link in the left menu, you will be redirected to the application registration page. You have to enter the following details:

- **Application Name -** this is simply an identifying name of your web application used in the LiveID system and has no effect on functionality
- **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 site entered in the Domain name field
- **Domain name** - enter the fully qualified domain name of your web site
- **Secret key** - enter some key that will be used to encrypt data transferred between your web site and the LiveID server; be sure to remember this key as you will have to enter it in Kentico CMS settings
- **Application verifier required** - it is recommended to choose 0



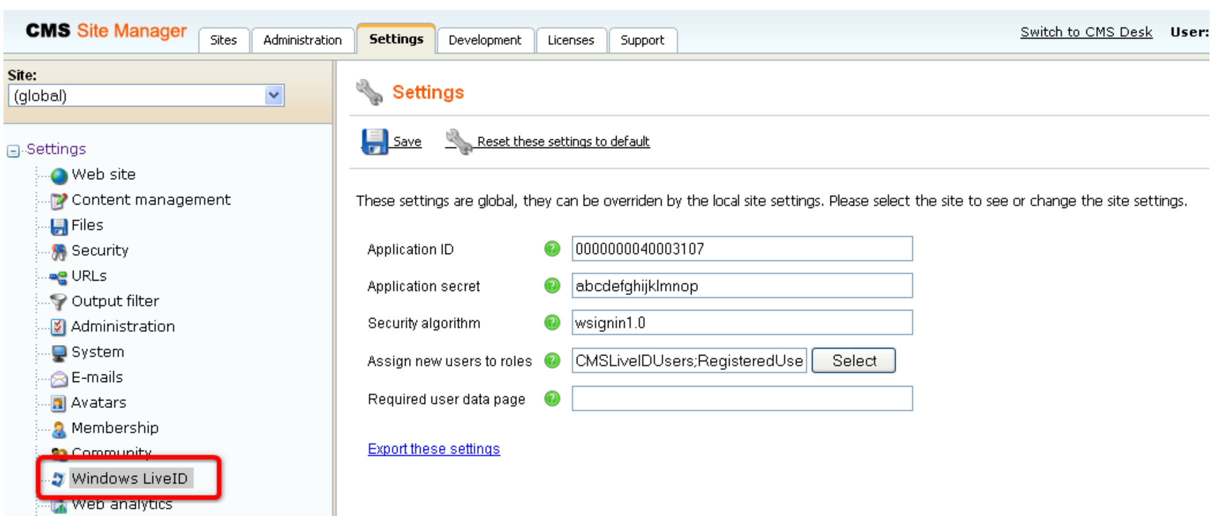
When you're finished, retype the CAPTCHA, agree with terms of use and click the **Submit** button. You will be displayed with the following message, showing you the **Application ID**. You will use this ID later in Kentico CMS settings.




## 22.6.3 Settings

LiveID 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 at the top left part of the page.

- **Application ID** - identifier of your web site; you were given this ID when registering your web site to LiveID
- **Application secret** - secret code that will be used for encryption of messages transferred between your web site and the LiveID server; you entered this key when registering your web site to LiveID
- **Security algorithm** - algorithm used for encryption of messages transferred between your web site 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 these roles
- **Required user data page** - alias path to a page containing the 'Required LiveID user data' web part; if entered, then when a new LiveID user logs in to the site, her user account is not created automatically, but she is required to enter some additional data (or merge with an existing account) using the web part



## 22.6.4 Available web parts

After registering your web at msm.live.com/app and making the necessary settings, you can use the following two web parts on your website.

### Windows LiveID

This webpart can be used by site visitors to sign in to your web site using their Live ID. It can be placed on any page of your web site. 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.

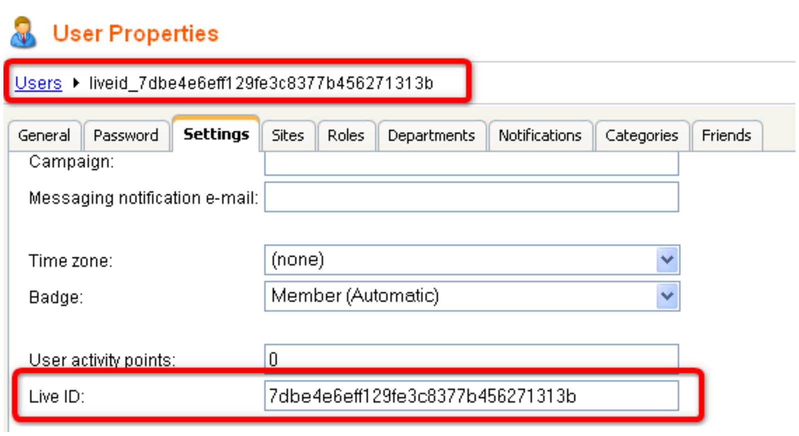

Although the web part works fine without setting any of its properties, the following can be set:

- **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

When some user signs in through LiveID for the first time, Kentico CMS automatically creates a new user account for her. Such account has the **User name** field filled, in format '*liveid_<liveidtoken>*'. The **LiveID** field contains the bare LiveID Token.

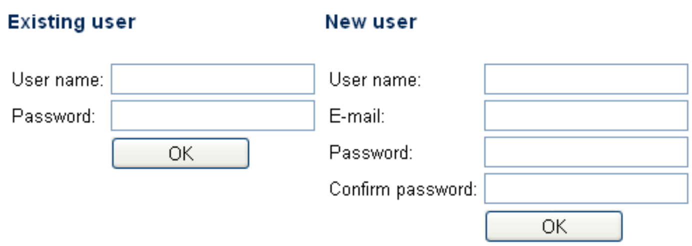
The LiveID Token is a hexadecimal identifier that KenticoCMS recognizes the user accessing the site via LiveID by. It is received from the LiveID server and is unique for each application - other web sites supporting LiveID logon will recognize the same user under some different LiveID Token.




In some cases, you may want the new users to provide some extra details before creating the new 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**.

In the left part, an existing site user can merge her current user account with the LiveID by just entering her user name and password. In this case, the LiveID Token will be added to the LiveID field of the existing profile.

New users can enter the required details in the right part of the web part.




The web part has the following specific properties:

- **Can set password** - if checked, new users will be able to set a password for their new account so that they can log in the usual way as well as via 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 not after LiveID logon (e.g. when accessed by entering its URL into the browser)

# 22.7 Badges

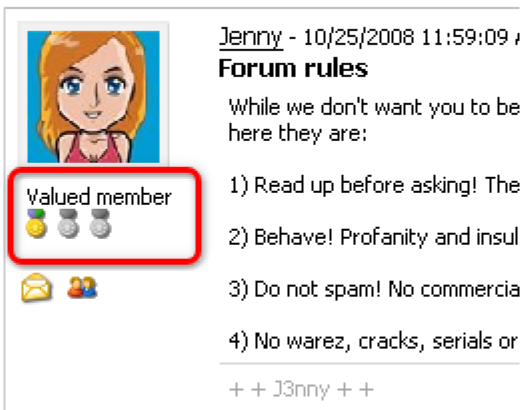
## 22.7.1 Badges

Users can be labeled with badges. These are images with a short text, expressing user's activity level, importance or role in the context of the web site. They are displayed in forum posts, on users' public profile or in your own custom control.

There are two types of badges:

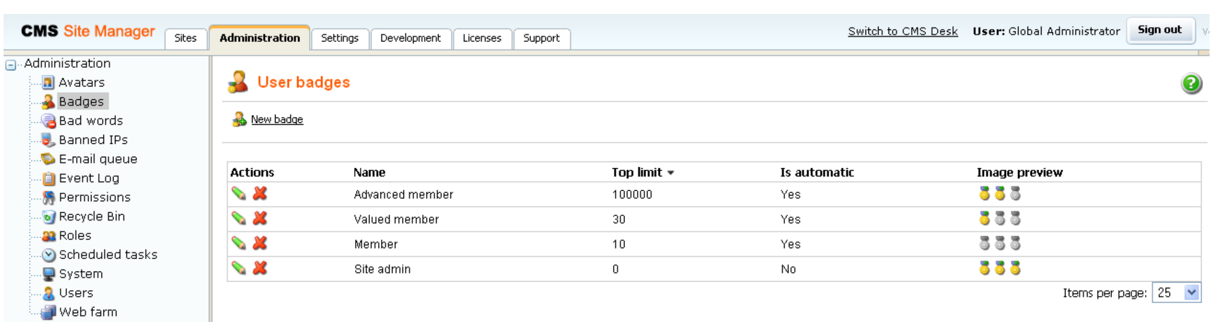
- **Automatic badges** - these are assigned to users based on the number of gained activity points
- **Non-automatic badges** - these are assigned to users manually by site administrators and are assigned permanently, regardless of the number of gained activity points

In the screenshot below, you can see one of the pre-defined badges in a forum post.




## 22.7.2 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

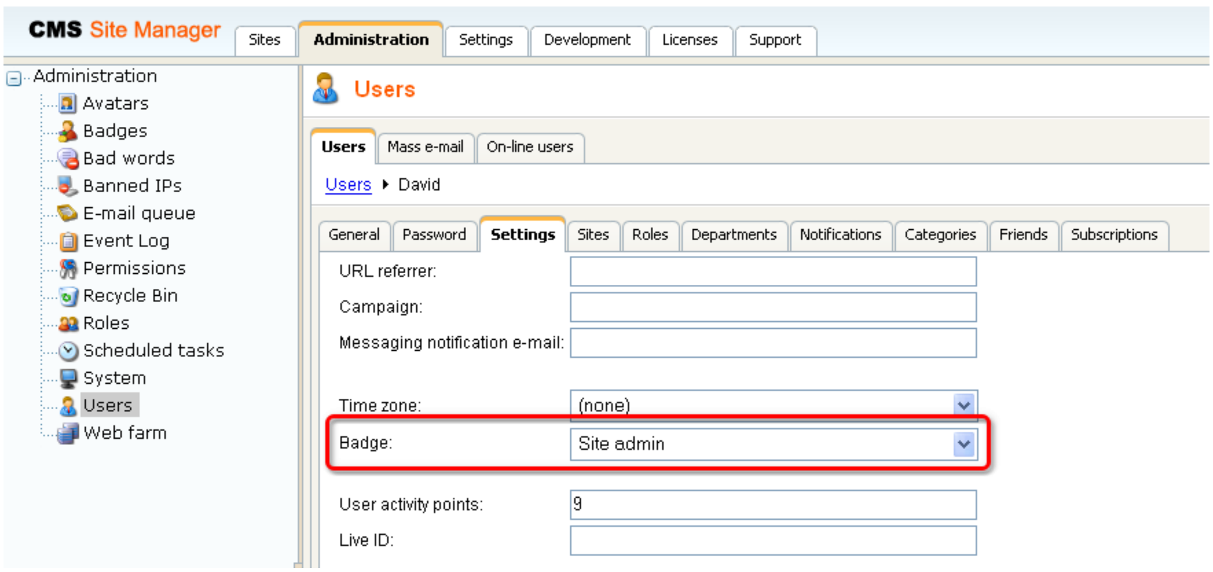
will remain assigned permanently, regardless of the number of gained activity points

- **Top limit** - number of activity points required for the user to get the badge; applies only to automatic badges

## 22.7.3 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.

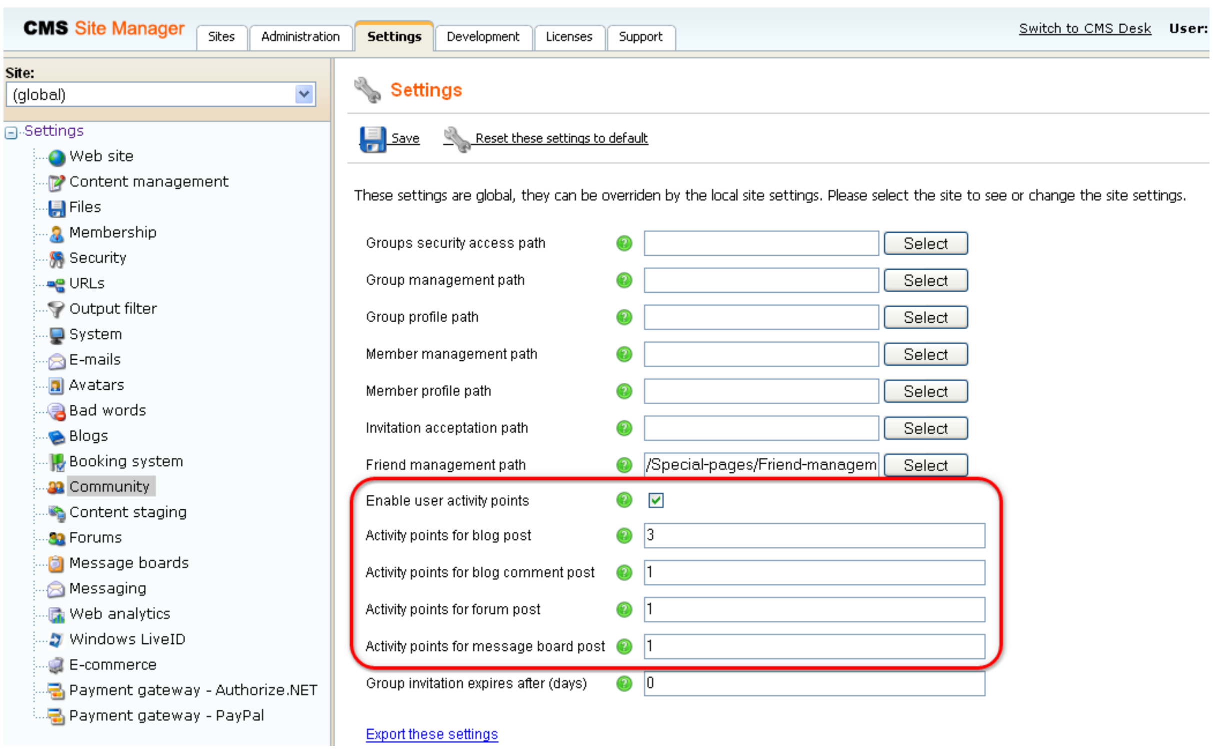


## 22.7.4 Activity points

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




## 22.7.5 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

# Part XXII

# 23 Microsoft Silverlight

## 23.1 Introduction

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, install banner leading to download link will be offered automatically.

You can find an example of a Silverlight application of the sample **E-commerce** site, in the **Silverlight** section.


### How it works in general:

1. Developer creates a web site with a built-in Silverlight application.
2. 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, she is 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 powerfull tool for designers **- Expression Blend**, which enables you to create application design in a really comfortable way. We also strongly recommend installing **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. Moreover, there you can find valuable tutorials which can help you get started developing Silverlight applications.

We also recommend reading the official Microsoft Silverlight documentation.

## 23.2 Adding Silverlight application to your site

To add your 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 sample **E-commerce site**, in the **Silverlight** section.

The following steps need to be taken to add your Silverlight application to a web site.

1) Go to **CMS Desk** and choose the page where you want to add your Silverlight application from the content tree.

2) Switch to the **Design** mode and click the **Add web part** () icon at the top-right 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** - minimal required version of Silverlight required by the silverlight application 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%*)
- **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 web site.

## 23.3 IIS configuration

Silverlight introduces two new file extensions:

- **.xaml** for XAML files
- **.xap** for the zip-based binary packaging format

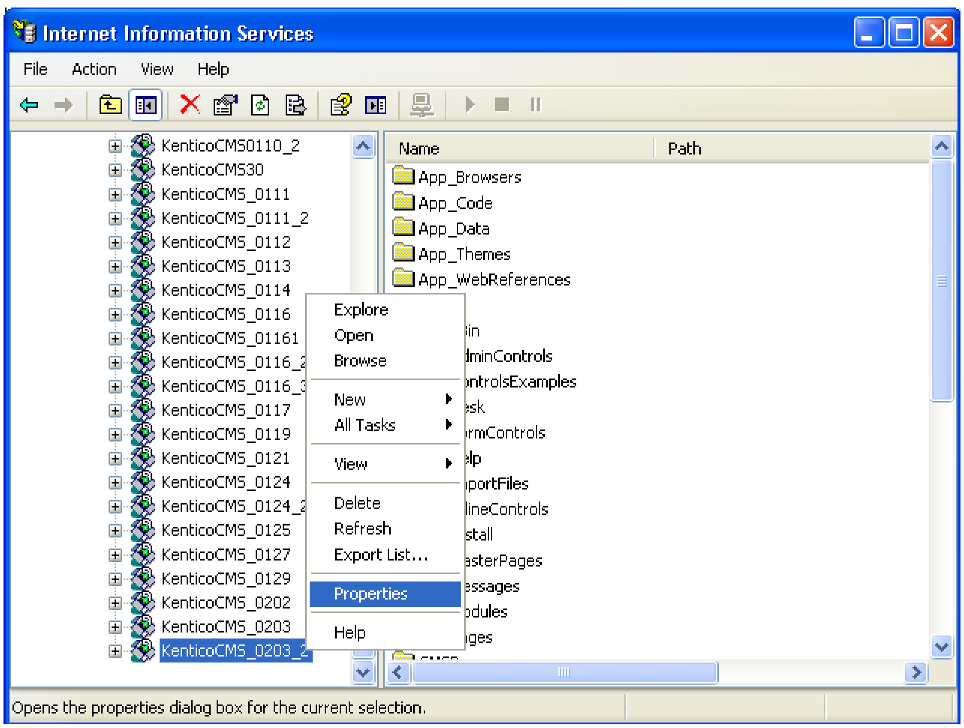
For your application to work correctly, you need to add the MIME types for these file extensions to your web server so that it recognizes Silverlight content appropriately:

- **Extension**: .xaml
- **MIME type**: application/xaml+xml

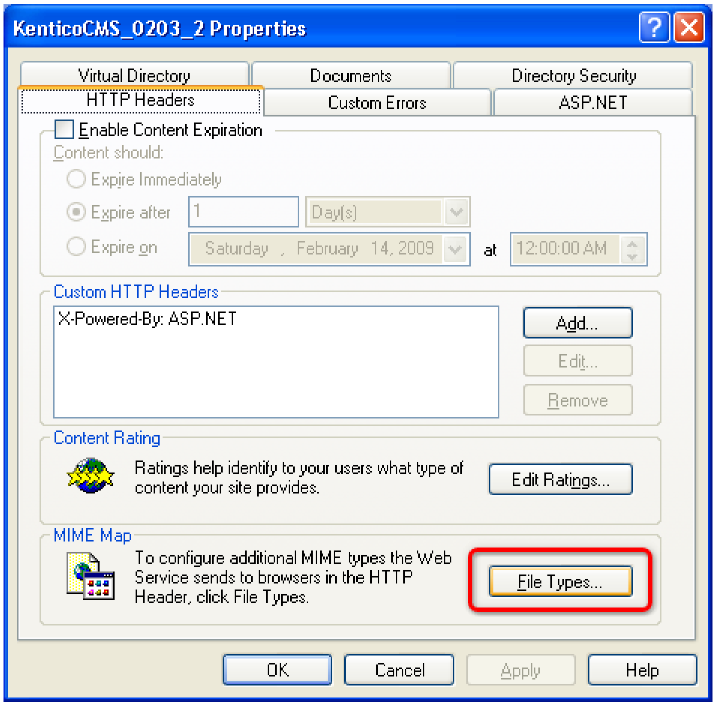
- **Extension**: .xap
- **MIME type**: application/x-silverlight-app

Here is a step-by-step guide on how to do it on IIS 5.1:

1) Open IIS, right-click your web site and choose **Properties**.



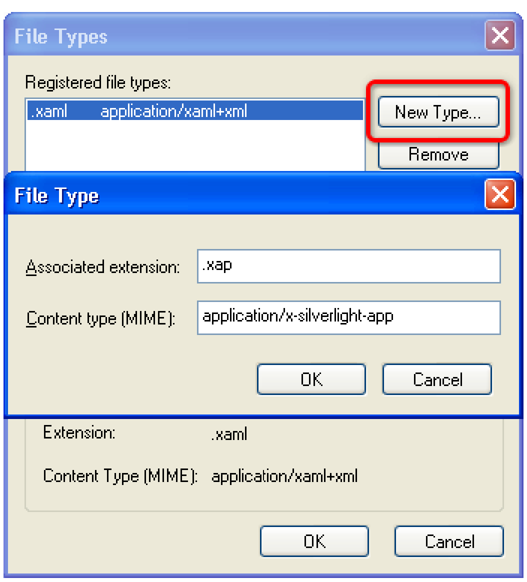
2) Switch to the **HTTP Headers** tab and choose **File Types** in the **MIME Map** section.




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



# Part XIV

# 24 Module Abuse report

## 24.1 Abuse report module overview

The Abuse report module enables site visitors to report indecent posts, comments or some other forms of web site abuse. The module comes with two web parts, Abuse report and In-line abuse report, enabling site visitors to send the reports.

## 24.2 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 that 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.

If you click the title of a report, you will be redirected to the page that the report was sent from. If you mouse-over it, the report description entered by the sender of the report will be displayed in form of a tooltip.

The following actions are available for each of the 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
- **Reject ( ) -** switches the report to the Rejected status; used to mark reports that were not considered being cases of web site abuse

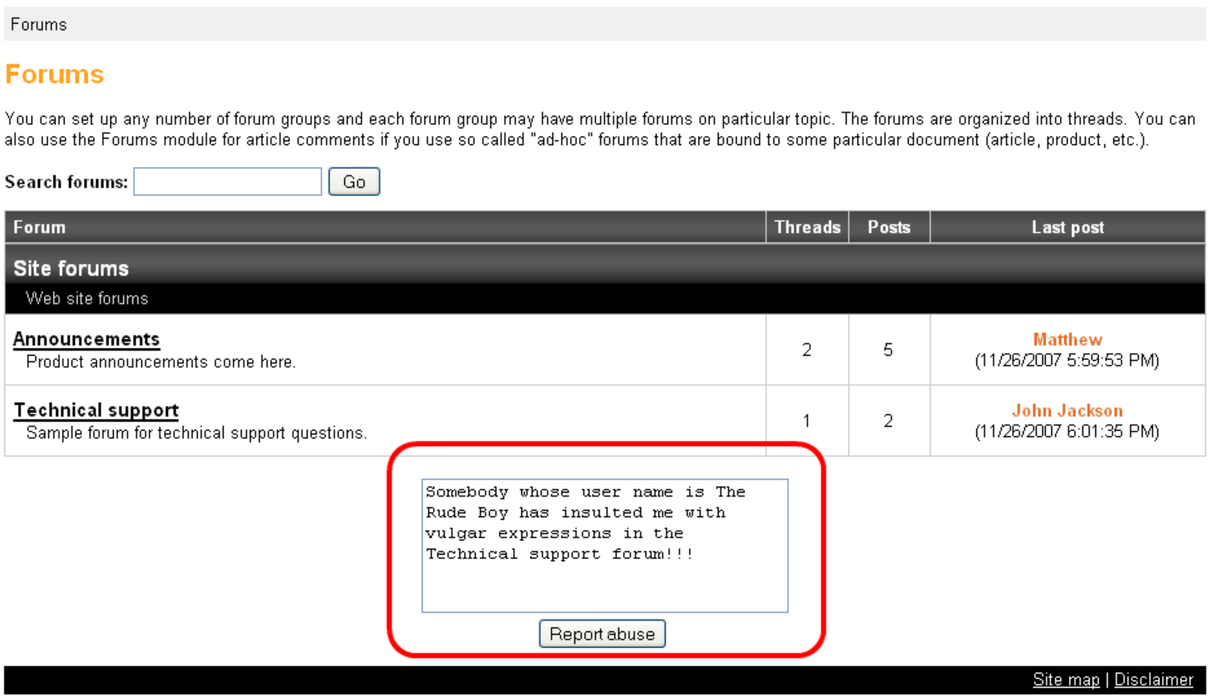
## 24.3 Using the Abuse report and In-line abuse report web parts

The **Abuse report** web part serves site visitors to report web site abuse such as using indecent expressions, offensive posts, etc. The web part can be placed into any web part zone on any page of your web site. In the web part selection dialog, you can find it under the **Abuse report** folder.

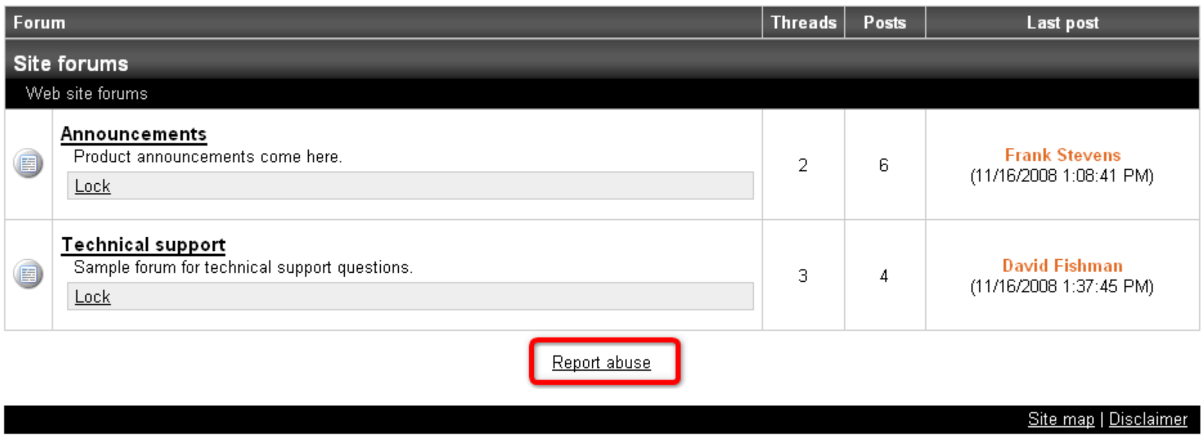
You have to set the following specific properties in order for the web part to work properly:

| | |
|---|---|
| Confirmation text | Text message that will be displayed to the site visitor after sending the abuse report. |
| Title of abuse report | Title of the report that will be displayed to the site administrator in the list of abuse reports. |

When the user types in the report text and clicks the **Report abuse** button, a report will be logged in **CMS Desk -> Tools -> Abuse report**.



If you want to use only a small clickable link instead of the whole **Abuse report** web part, you can use the **In-line abuse report** web part. When  a site visitor clicks the link, the page will become grayed-out and an abuse report dialog will pop-up in the middle of it. Properties of the two web parts are identical. This web part can also be used in transformations, as shown in chapter Using the In-line abuse report web part in transformations.




## 24.4 Using the In-line abuse report web part in transformations

The Inline abuse report web part appears as a link with the text 'Report abuse'. After clicking the link, a dialog appears, letting the site visitor send abuse report to the site administrators. This web part is mainly used in transformations, letting you include the Report abuse link repeatedly where ever you need.

The object viewer that is being used by this web part can handle only blog comments, message board messages and forum posts. If you use the web part in transformations of some other objects, the "Show object details" link will not be displayed in its abuse reports' properties.

In the following example, you will learn how to add the Inline abuse report web part to the default transformation of a message board message on our sample Corporate Site.

1) Go to **CMS Desk**, switch to the **Edit** mode and its **Design** tab.

2) In the content tree, select **Examples -> Webparts -> Message boards -> Message board**.

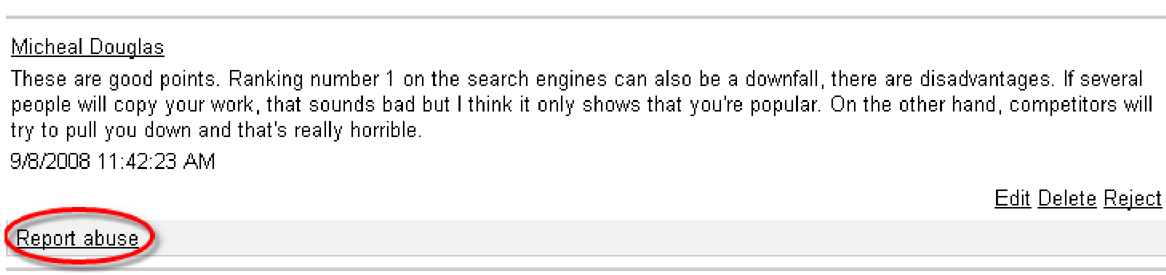
3) Choose to **Configure** () the **MessageBoard** web part.

4) In the web part properties window, choose to **Edit** the **Message transformation** property.

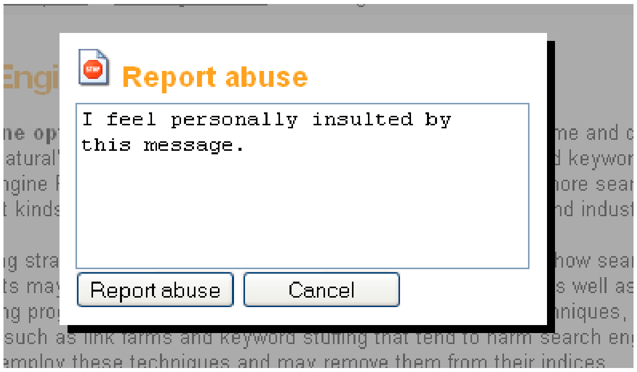
5) Replace the transformation code with the following code. It is the original code of the transformation with the highlighted parts added. The first highlighted part is registration of the webpart tag, the second one is a <div> containing the web part itself:

```
<%@ Register Src="~/CMSModules/MessageBoards/Controls/MessageActions.ascx"
TagName="MessageActions" TagPrefix="cms" %>

<%@ Register Src="~/CMSModules/AbuseReport/Controls/InlineAbuseReport.ascx"
TagName="InlineAbuseReport" TagPrefix="cms" %>

<div class="CommentDetail">
    <asp:Panel ID="pnlRating" runat="server" />
    <table width="100%">
        <tr>
            <td class="CommentUserName" style="width: 100%">
               <%# IfEmpty(Eval("MessageURL"), Eval("MessageUserName"), "<a href=\"" +
Eval("MessageURL") + "\" target=\"_blank\">" + Eval("MessageUserName") + "</a>")%>

            </td>
        </tr>
        <tr>
            <td class="CommentText">
               <%# CMS.GlobalHelper.TextHelper.EnsureLineEndings(Convert.ToString(Eval
("MessageText")), "<br />")%>
            </td>
        </tr>
        <tr>
            <td class="CommentDate">
               <%# GetDateTime(Eval("MessageInserted")) %>
            </td>
        </tr>
        <tr>
            <td align="right" class="CommentAction">
                <div class="buttonpedding">
                      <cms:MessageActions ID="messageActions" runat="server" />
                </div>
            </td>
        </tr>
    </table>
</div>

<div class="BlogPDateWhole">
<cms:InlineAbuseReport ID="InlineAbuseReport" runat="server" ReportObjectType="board.
message" ReportObjectID='<%# Eval("MessageID") %>'  ReportTitle="Message board abuse
report" />
</div>

<hr style="border: 1px solid #CCCCCC;"/>
```

6) Click **Save**. Now if you switch to the Live site and view the message board, you should be able to see the Abuse report link below each of the messages:

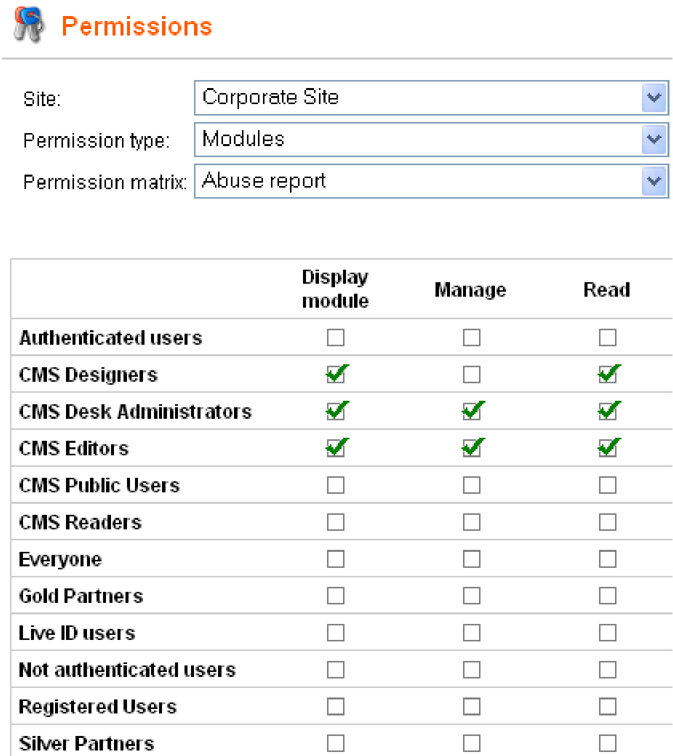


7) If you click it, the screen will become grayed out and the following dialog will pop up in the middle of it, as you can see in the screenshot below. Type in some report text and click the **Report abuse** button. The report will be logged in **CMS Desk -> Tools -> Abuse report**.



## 24.5 Security

Permissions for the Abuse report module can be set in **Site Manager -> Administration -> Permissions**. Select the **Modules -> Bad words** permission matrix and grant user roles with appropriate permissions.



Permissions

Site: Corporate Site

Permission type: Modules

Permission matrix: Abuse report

| | Display module | Manage | Read |
|---|---|---|---|
| Authenticated users | ☐ | ☐ | ☐ |
| CMS Designers | ☑ | ☐ | ☑ |
| CMS Desk Administrators | ☑ | ☑ | ☑ |
| CMS Editors | ☑ | ☑ | ☑ |
| CMS Public Users | ☐ | ☐ | ☐ |
| CMS Readers | ☐ | ☐ | ☐ |
| Everyone | ☐ | ☐ | ☐ |
| Gold Partners | ☐ | ☐ | ☐ |
| Live ID users | ☐ | ☐ | ☐ |
| Not authenticated users | ☐ | ☐ | ☐ |
| Registered Users | ☐ | ☐ | ☐ |
| Silver Partners | ☐ | ☐ | ☐ |

The following table explains the particular permissions:

| | |
|---|---|
| Display module | The module is visible in the administration interface. |
| Manage | The user is allowed to manage abuse reports. |
| Read | The user is allowed to view the abuse reports list. |

# Part XXV

# 25 Module Alternative forms

## 25.1 Alternative forms module 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.

## 25.2 Creating an alternative form

This example shows how to create an alternative form to the existing **Contact us** BizForm on the sample Corporate Site. Alternatives to document types, system tables and custom tables can be created in exactly the same way as described here. The user interface for doing this is found in the following locations:

- **Site manager -> Development -> Custom tables**
- **Site manager -> Development -> Document types**
- **Site manager -> Development -> System tables**

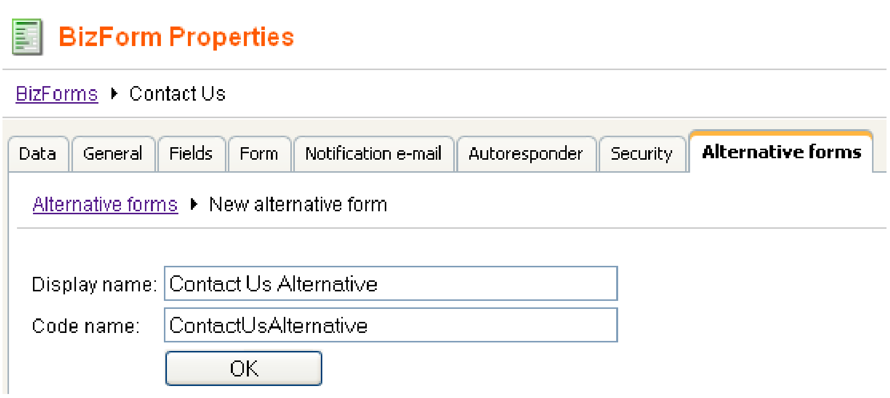
There, just choose to **Edit** ( ) the particular item, switch to its **Alternative forms** tab and follow the instructions below from the second step.

1) Go to **CMS Desk -> Tools -> BizForms**. Choose to **Edit** ( ) the **Contact us** BizForm. The **BizForm Properties** screen will appear.

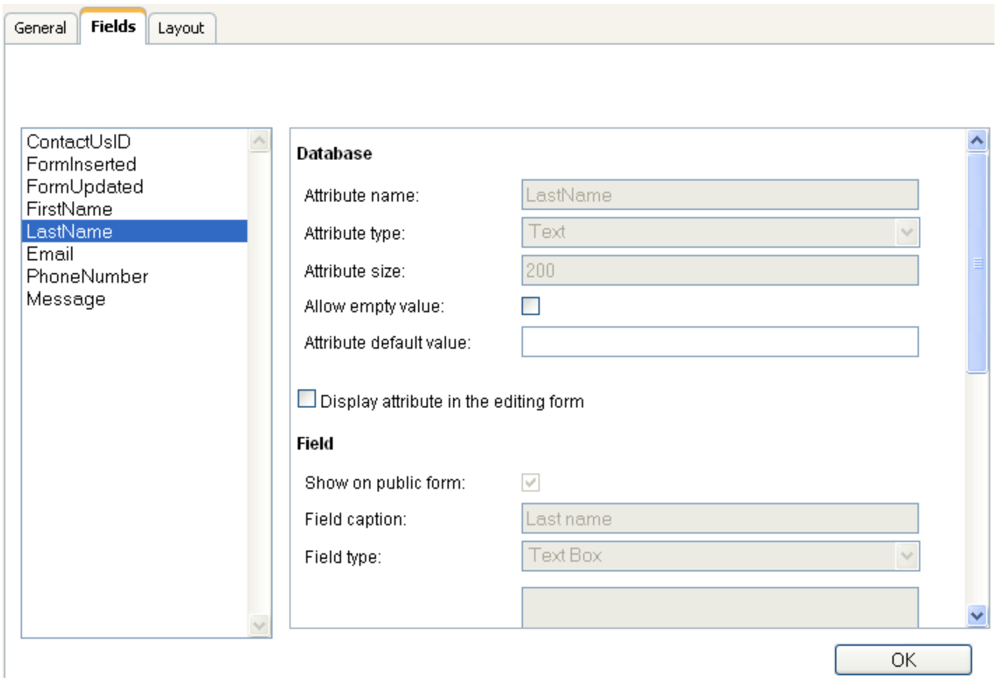
2) On **BizForm Properties**, switch to the **Alternative forms** tab and choose to **Create new form**.

3) On the form that will be displayed, enter the following details and click **OK**.

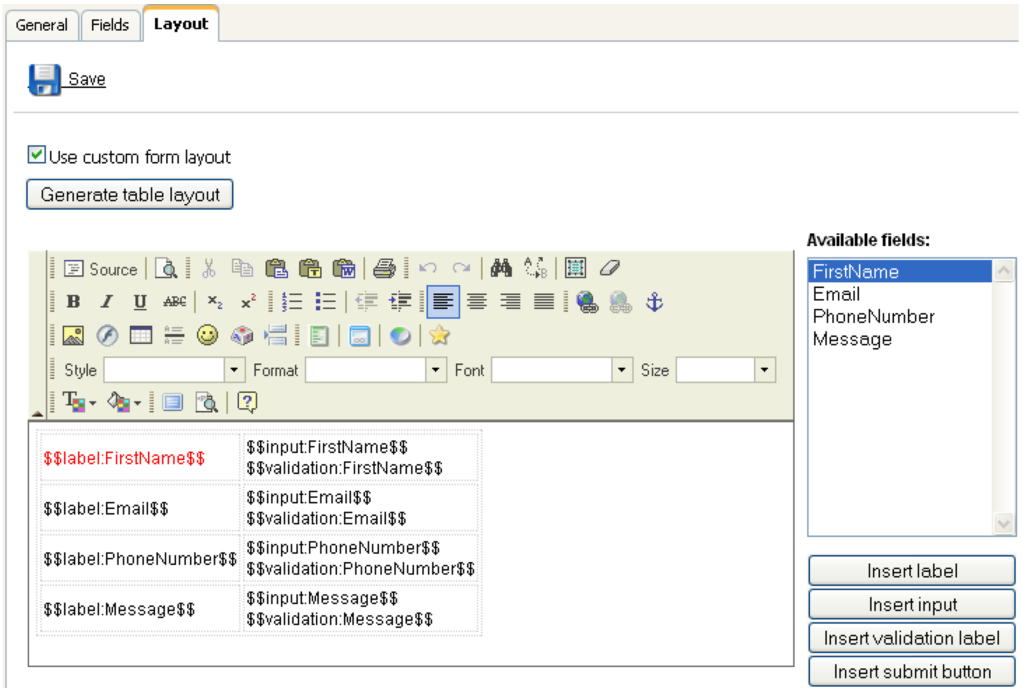
- **Display name:** Contact Us Alternative
- **Code name:** ContactUsAlternative



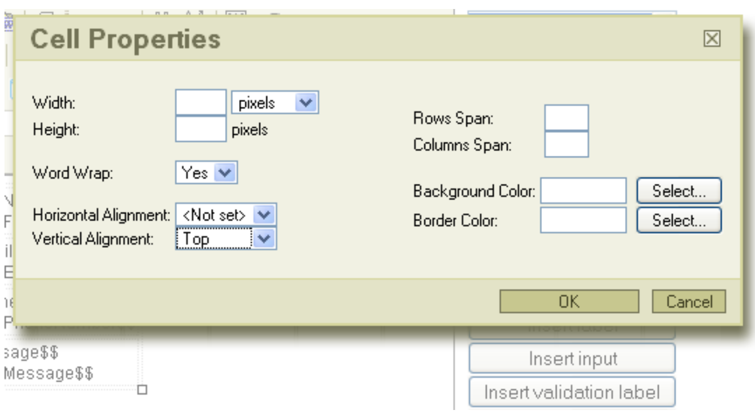
4) Switch to the **Fields** tab. As you can see, all fields present in the original form are present here as well and you can now make modifications to them. Let's disable the **LastName** field as an example. Select the field from the list on the left and on it's properties, uncheck the **Display attribute 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.

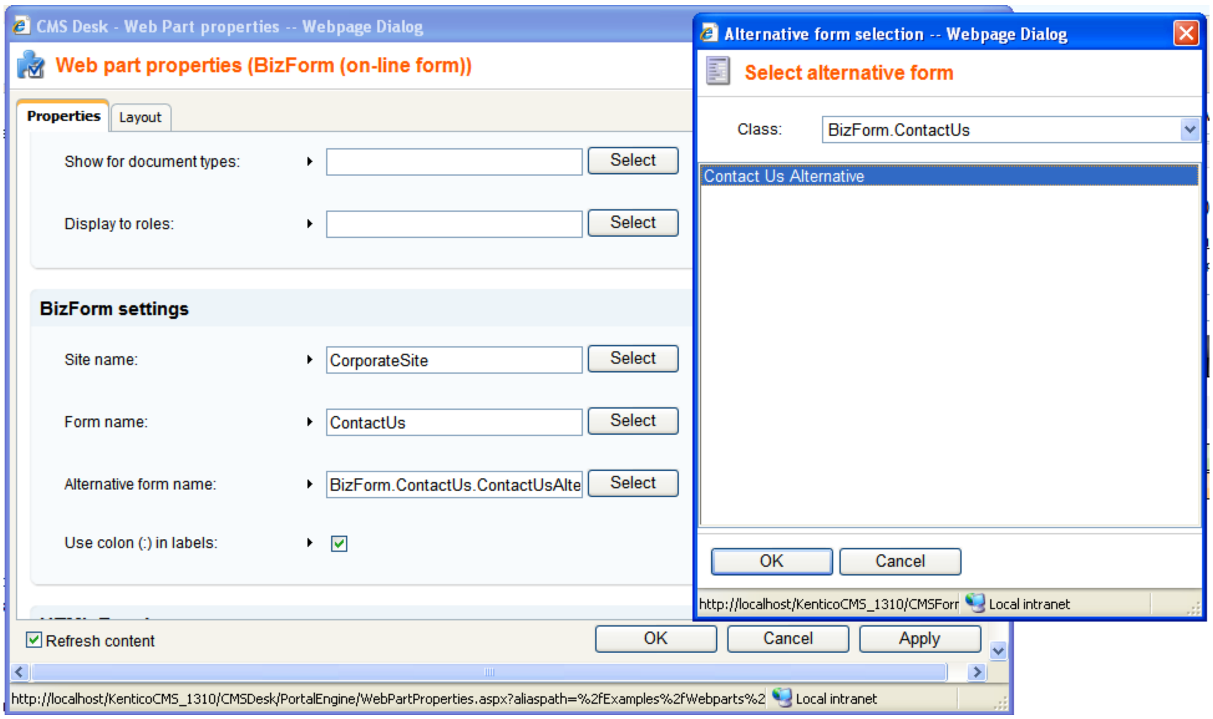



Also select the rows of the first table column, right click it and from the context menu, choose **Cell -> Cell properties**. In the displayed dialog, choose **Vertical alignment: Top** and click **OK**. This will make the labels to be aligned to their fields. Click **Save**.

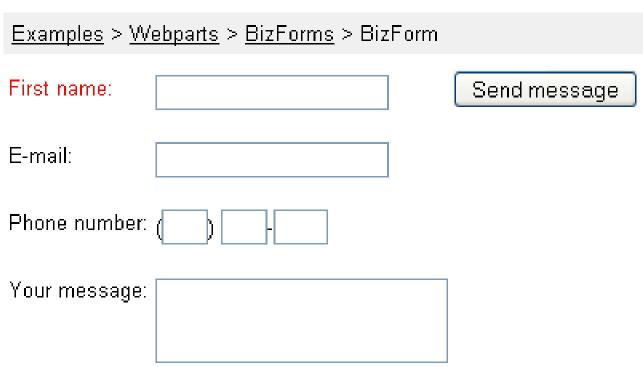


6) Let's take a look at what we've created. Switch to **Site Manager -> Content**. From the context tree, select **Examples -> BizForms -> BizForm**. As you can see, there is the original version of the **Contact Us** form present on the page. We will edit the web part's properties so that the alternative version will be displayed.

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**.



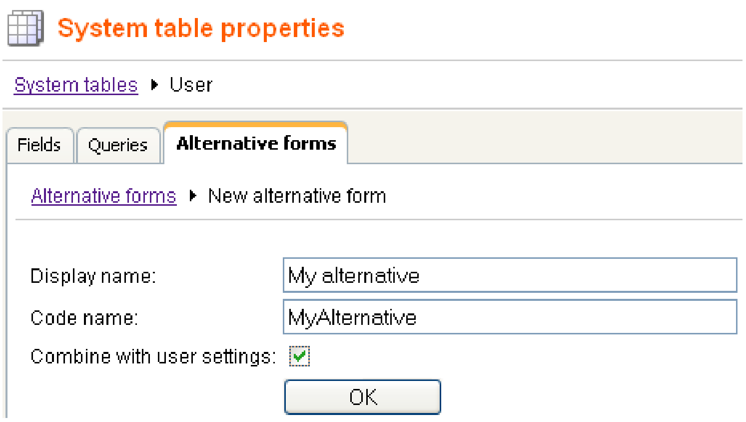

8) Now when you switch to the **Live site** mode, you should see the modified version as in the following screenshot:



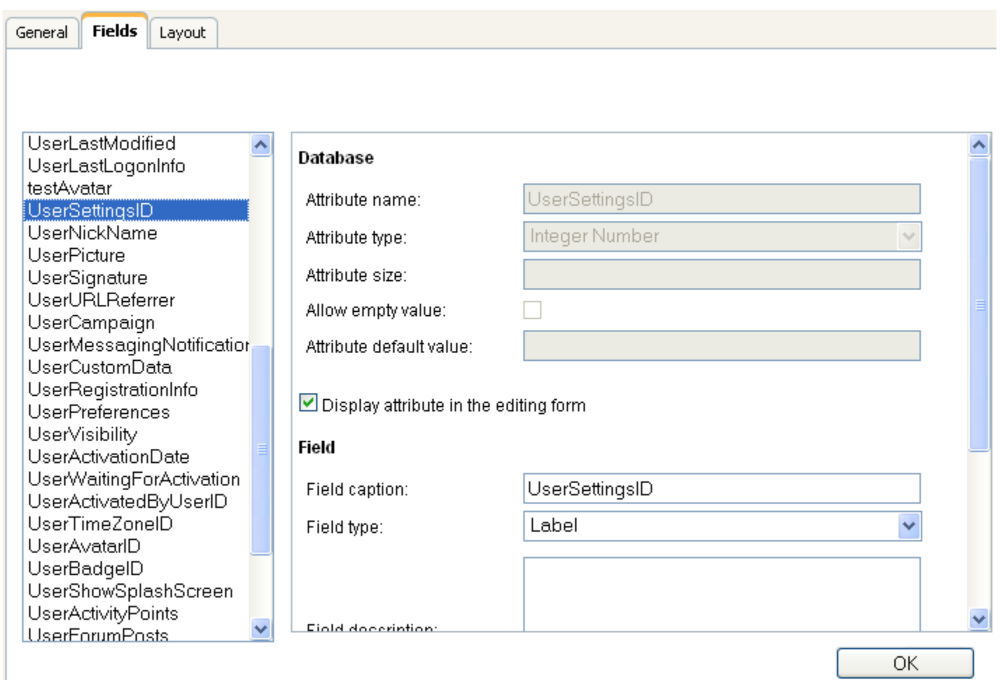

You have learned how to create an alternative form to an existing BizForm and use it on your web site.

## 25.3 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 table. When creating an alternative form to 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.




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:



## 25.4 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

| 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 |

# Part XXV

# 26 Module Avatars

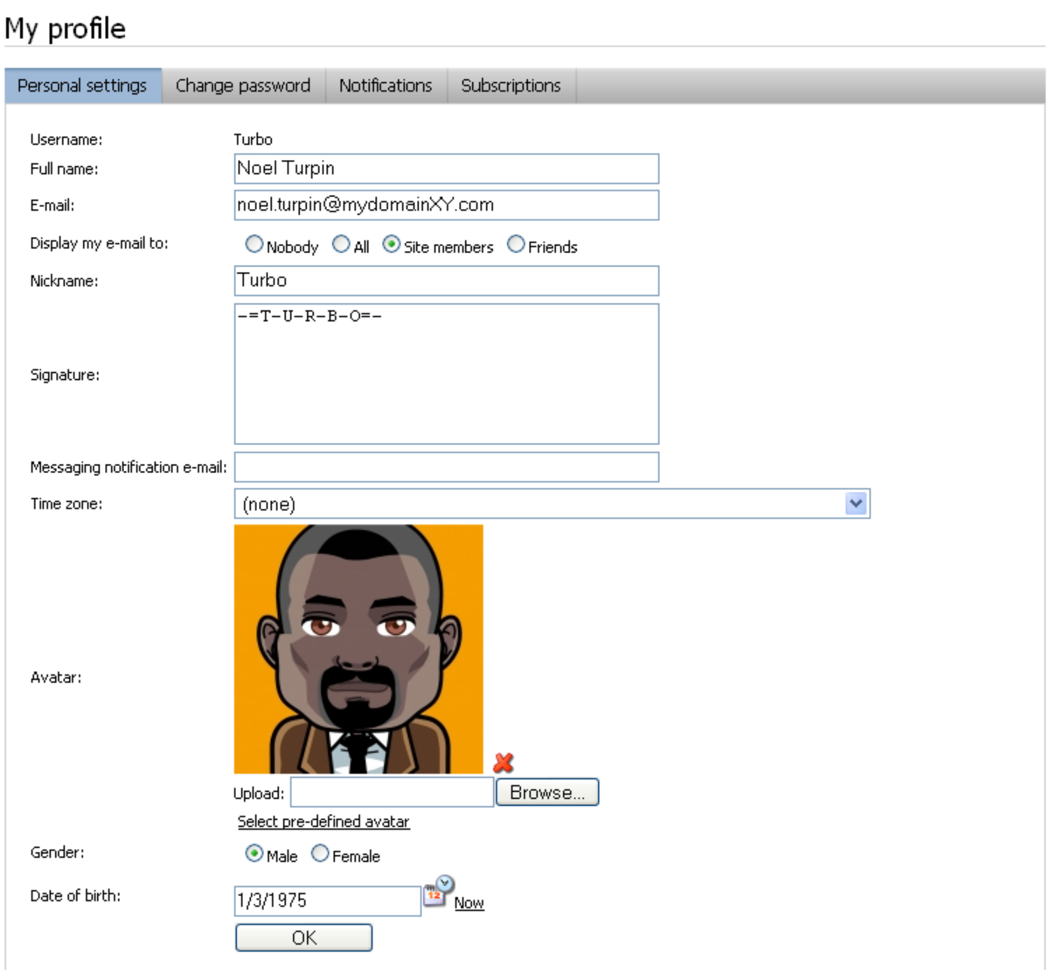
## 26.1 Overview

The Avatars module enables users to have an image associated with their account. This image is called 'avatar' and will be displayed on the user's public profile, in forum posts, etc. Users can either choose one from a gallery of pre-defined avatars or upload their own image from a file on their local disk.

Groups can also have avatars, while these will be displayed in the group's profile and can be beneficial to better and faster group identification, etc.

## 26.2 Changing user avatars

When a new user registers to a site, she will be assigned with the default avatar. After that, she can change her avatar using the **My account** web part. It can be done on its **Personal settings** tab, as you can see in the screenshot below.



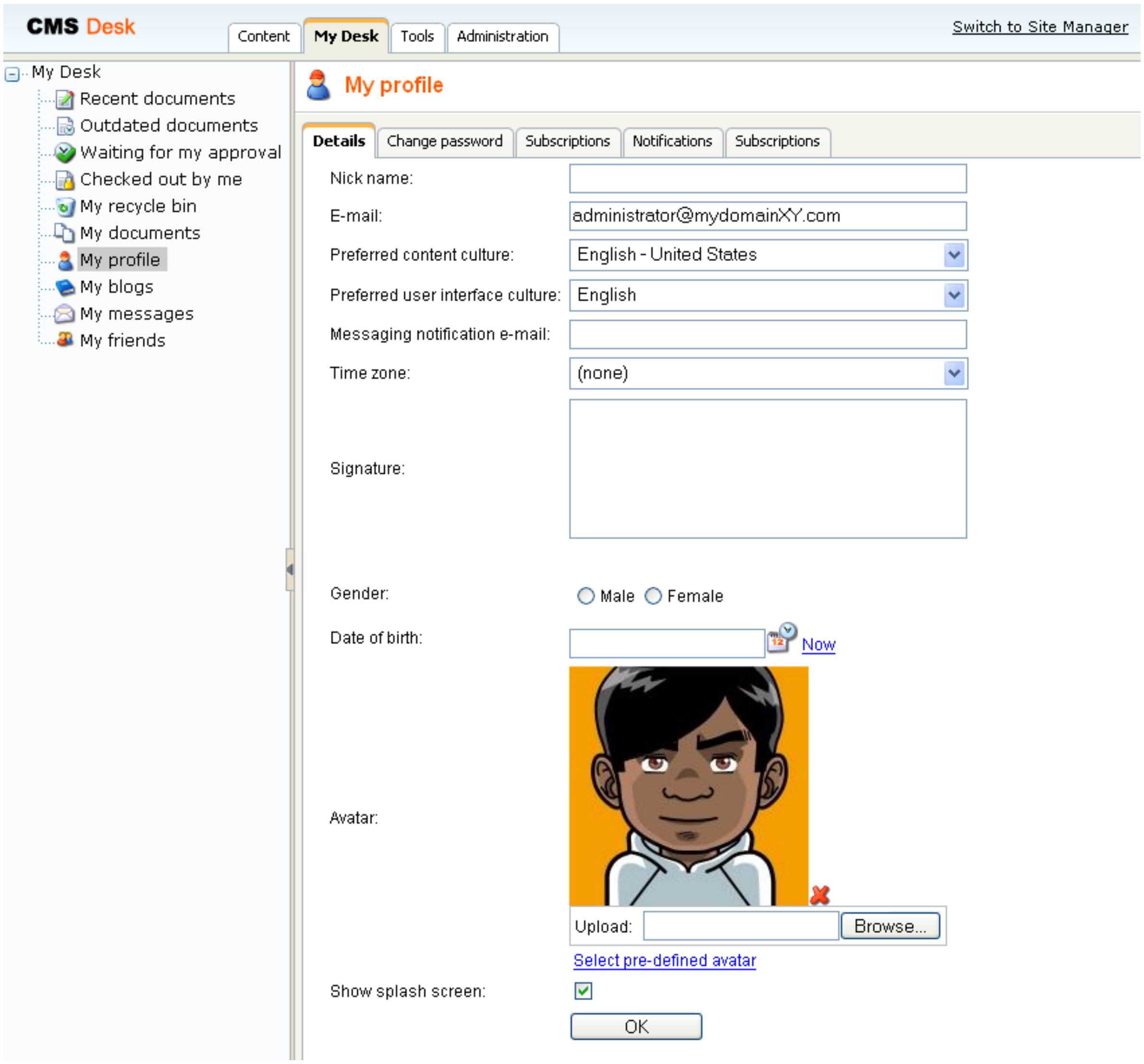

Users can **Delete** () the avatar or **Upload** a custom one from a file. If pre-defined avatars are enabled in site settings, users can also click the **Select pre-defined avatar** link, which displays a gallery of pre-defined avatars.


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.

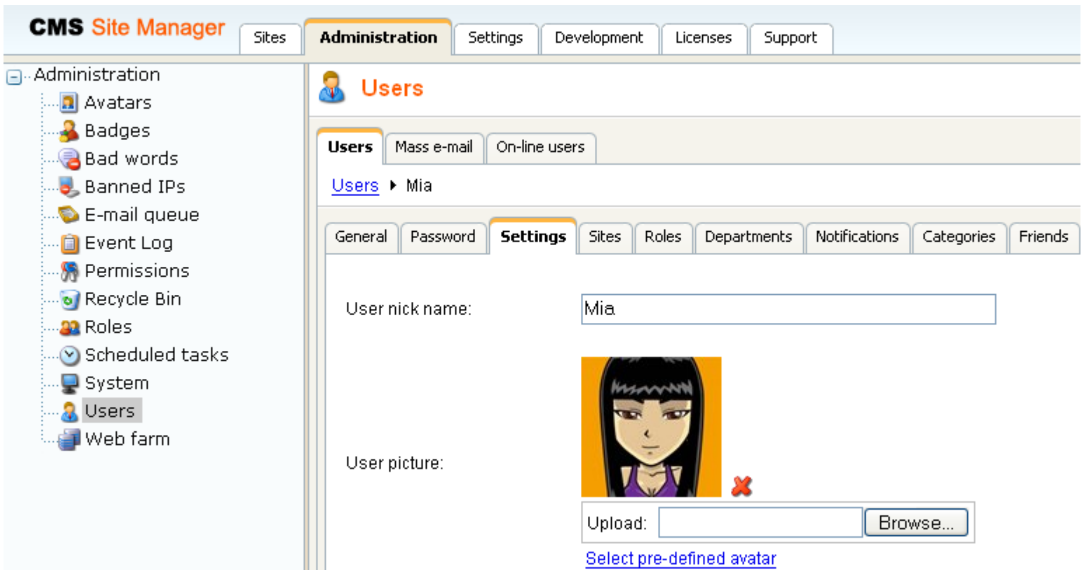
If default avatars are enabled in site settings, you can also click the **Select pre-defined avatar** link, which displays a gallery of pre-defined avatars from that you can easily pick one by clicking it and clicking **OK**.



## 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 her **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 that you can easily pick one by clicking it and clicking **OK**.

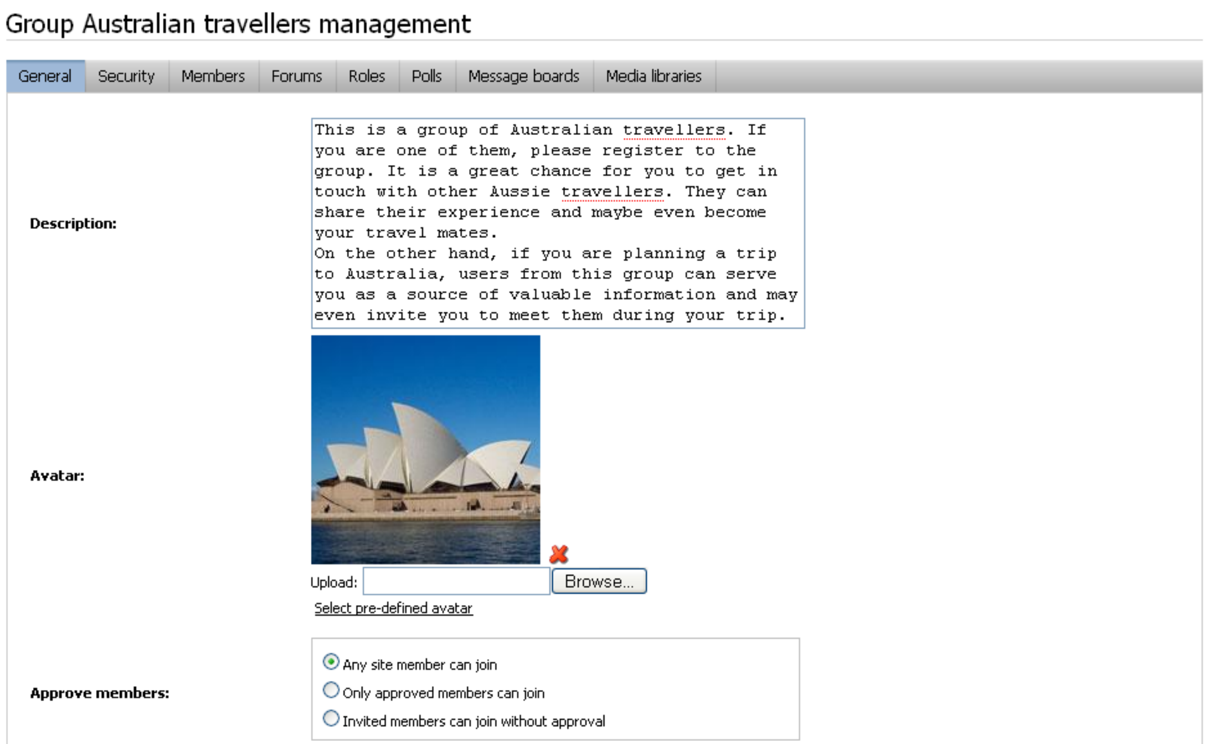


## 26.3 Changing group avatars

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.

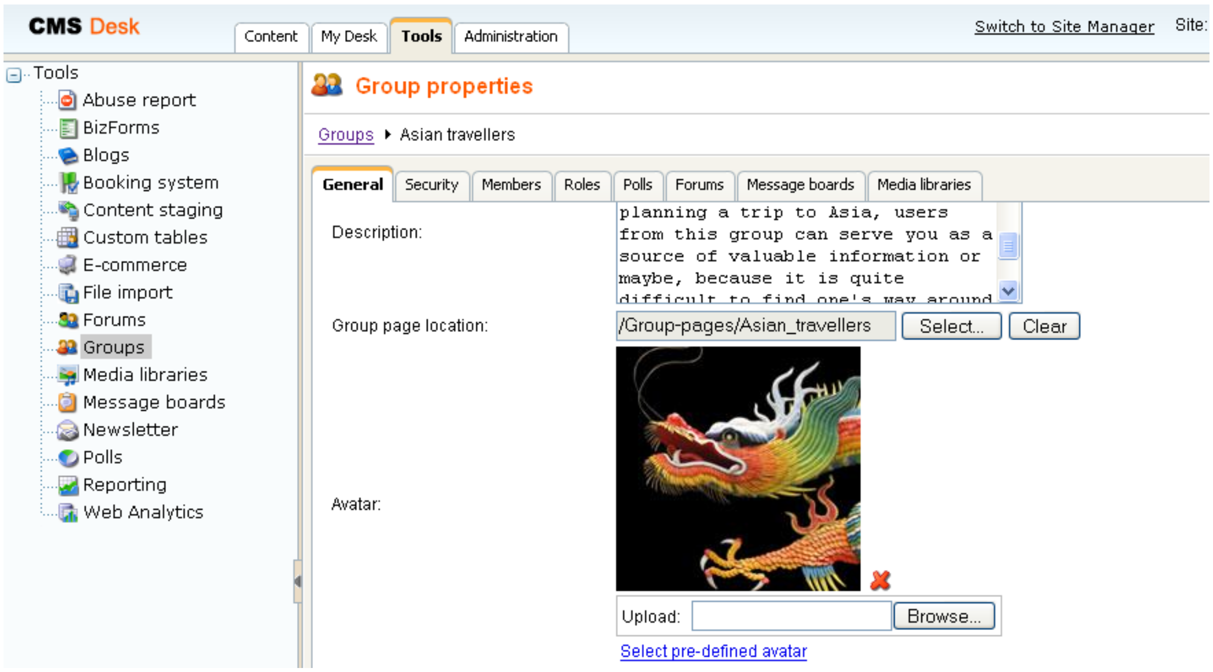
Users can **Delete** () the avatar or **Upload** a custom one from a file. If pre-defined avatars are enabled in site settings, users can also click the **Select pre-defined avatar** link, which displays a gallery of pre-defined avatars.

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 **Australian travellers** group in the **My groups** section. Click it, you will be displayed with what you can see in the screenshot below.



## 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. Group avatar can be **Deleted** ( ), new one can be **Uploaded** from a file or selected from a gallery of pre-defined avatars (in case that it is enabled in site settings).

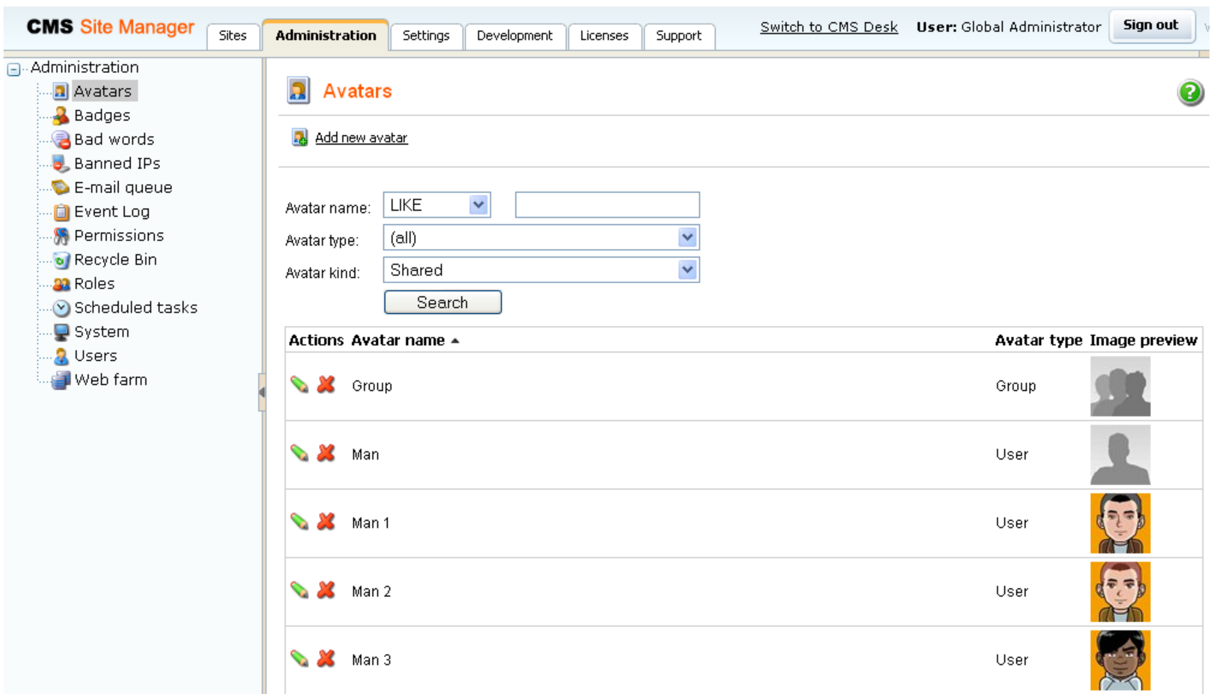


## 26.4 Managing avatars

The administration interface for avatars 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 **Add new avatar** link at the top part of the page. Click it.

2) You will be asked to enter the following details:

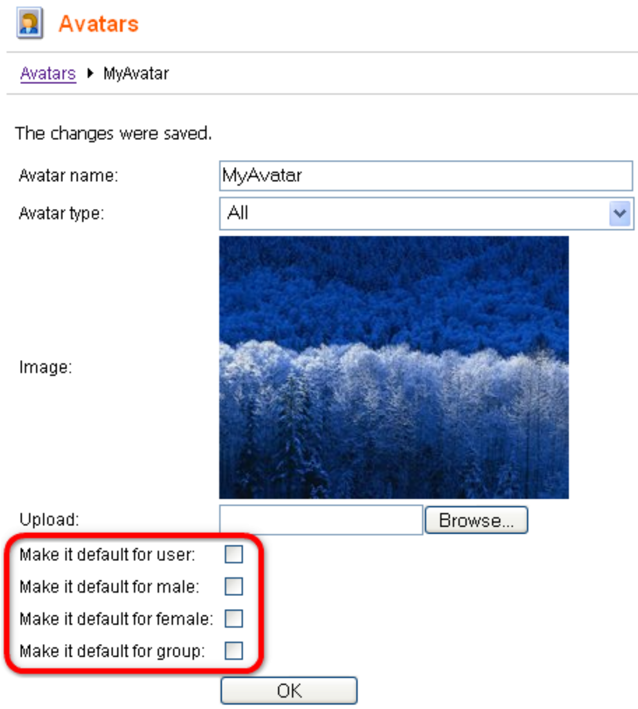
- **Avatar name** - name of the avatar
- **Avatar type** - choose if the avatar can be used for users, groups or both
- **Upload** - enter the path to the avatar image on your local machine or click the **Browse** button to browse and locate the file

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 a newly created users or groups automatically when the user or group is created.



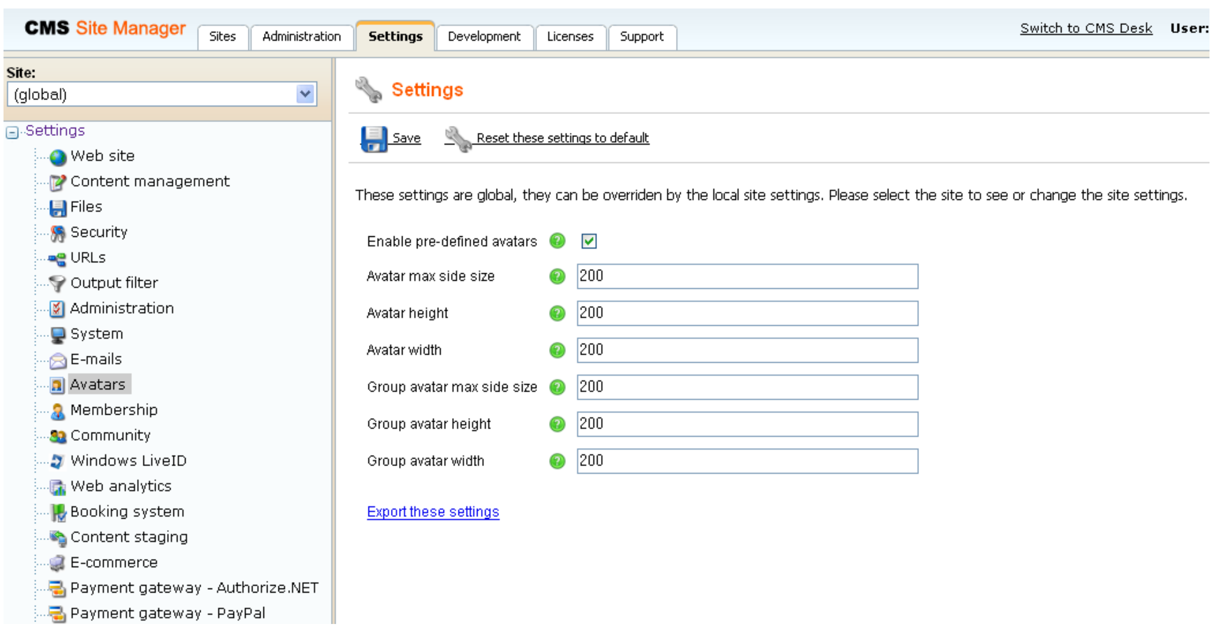
## 26.5 Settings

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 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



# Part XV

# 27 Module Bad words

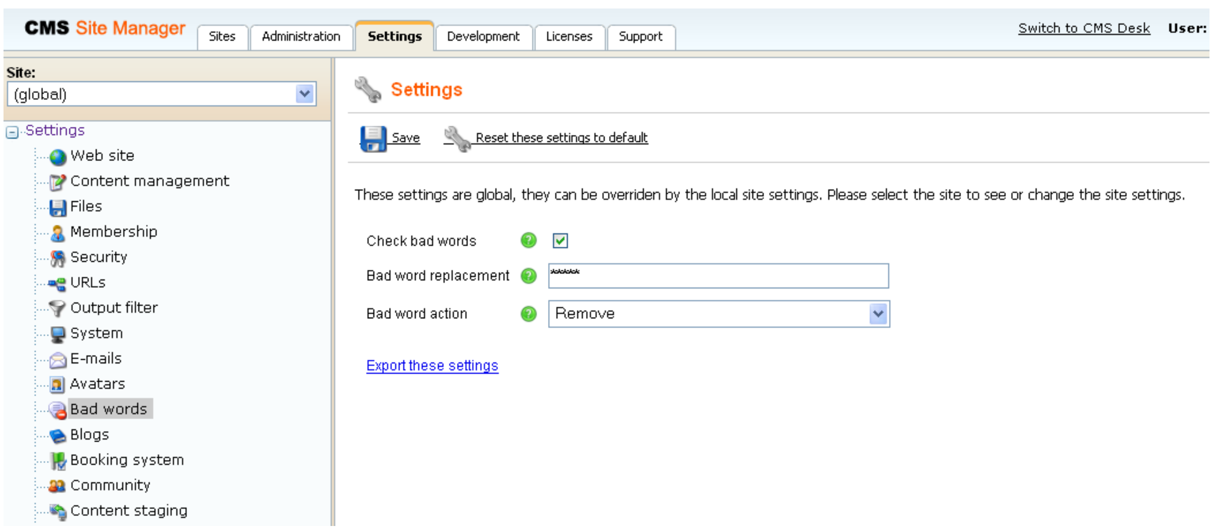
## 27.1 Bad words module overview

The Bad words module can be used as a filter for indecent input from web site users. The module can filter forum posts, blog comments, messages sent via the Messaging module and messages posted onto a message board. You can define an unlimited number of bad words. If a user inputs a text containing some of these words, a predefined action will be performed for the words.

## 27.2 Enabling the module

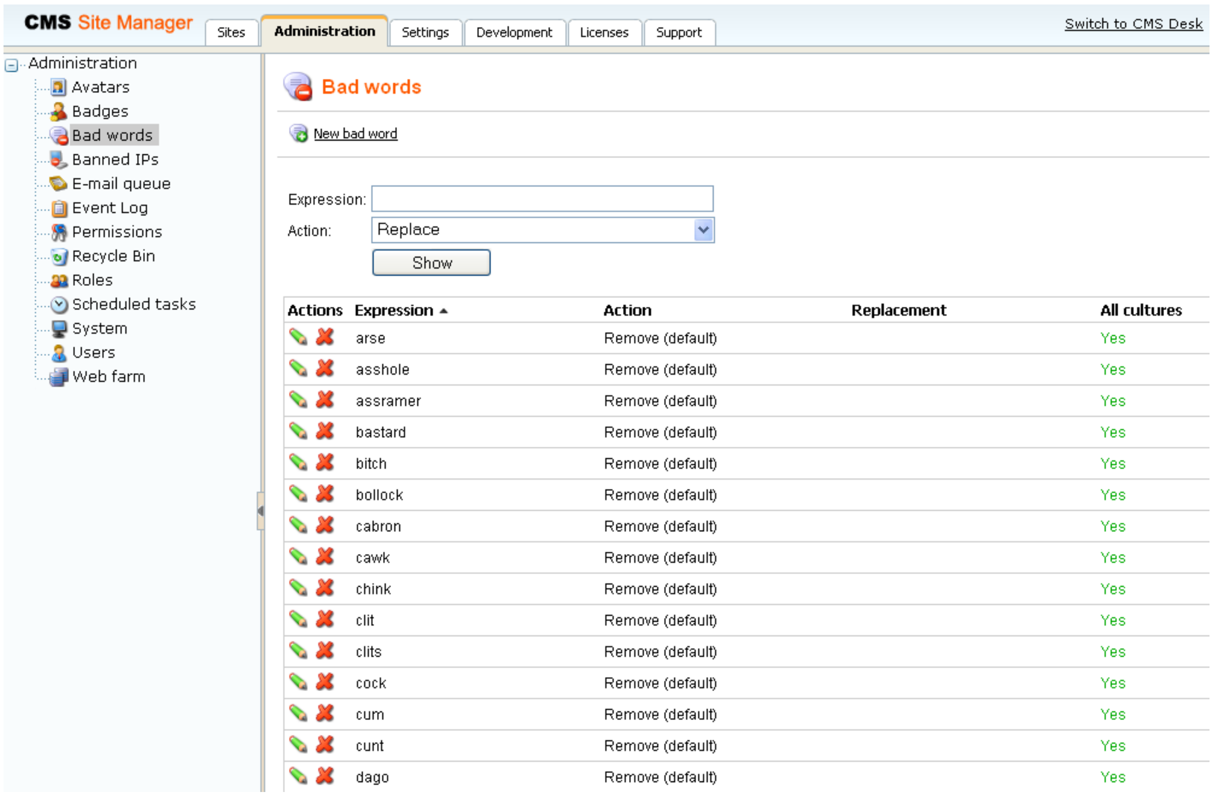
For the module to be functional, you also have to go to **Site Manager -> Settings -> Bad words** and check the **Check Bad Words** check-box.

You can also set the default action that will be taken in case that a bad word is detected and the word has no action defined. This is done using the **Bad word action** drop-down list. Using the **Bad word replacement** property, you can define the default replacement text that will be used in case that a bad word is detected, has the Replace action set and has no replacement text defined.



## 27.3 Defining a bad word

The Bad words module's user interface is located in **Site Manager -> Administration -> Bad words**.

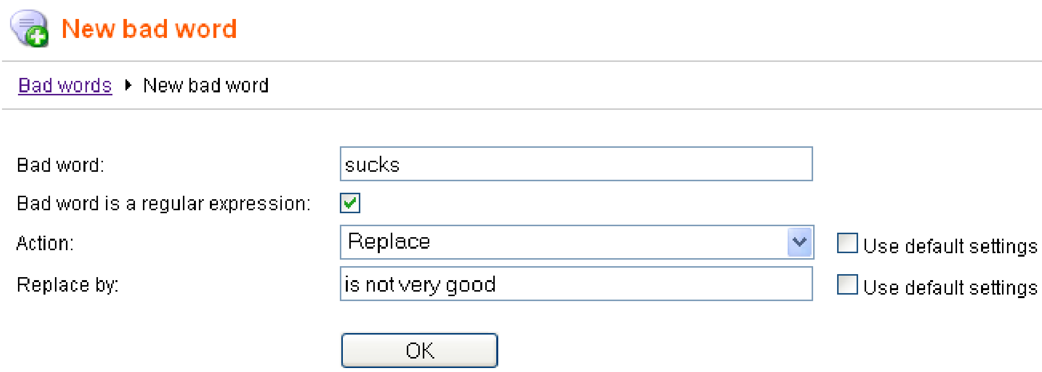



The top part of the page is a filter. Using it, you can display only those bad words that match the specified criteria. You can filter by the **Expression** and by the **Action** that has been set for it. After specifying the filtering criteria and clicking the **Show** button, only those items that match the specified criteria will be displayed in the list.

To add a new bad word, click the **New bad word** link at the top of the page. You will be redirected to the **Bad word properties** page. The following details can be entered:

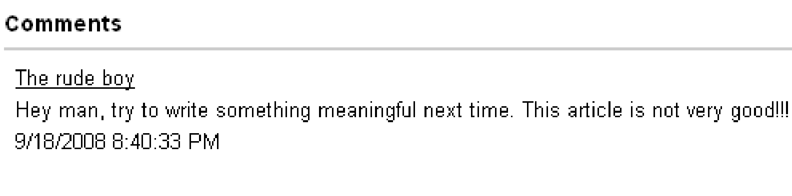
- **Bad word** - bad word that should not appear in the text
- **Bad word is a regular expression** - if checked, you can also enter regular expressions into the previous field
- **Action** - action that will be taken in case that the bad word is detected; see chapter **Possible actions** for more details
  Use default settings - if checked, global value will be used as set in ***Site Manager -> Settings -> Bad words -> Bad word action***
- **Replace by** - in case the Replace action is selected, the substitute for the bad word is defined here
  Use default settings - if checked, global value will be used as set in ***Site Manager -> Settings -> Bad words -> Bad word replacement***

Enter the details as in the following screenshot and click **OK**.

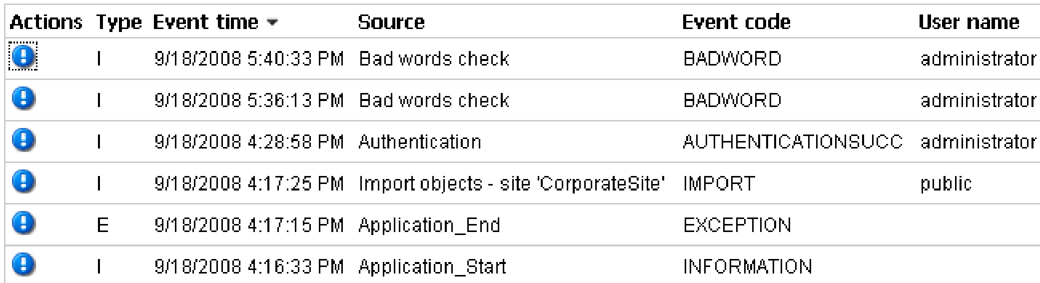



Let's try the functionality now. Go to the live Corporate Site, enter the Blog section and open some of the blog posts. Enter a comment as in the following screenshot and click **Add**.


As you can see, the last word has been replaced with its polite substitute that has been defined earlier.



You can also check the **Event log** in **Site Manager -> Administration**. An event is always logged automatically when a user tries to post some text containing a bad word. The **Event code** is **BADWORD** in such case.



| Actions | Type | Event time | Source | Event code | User name |
|---|---|---|---|---|---|
| | I | 9/18/2008 5:40:33 PM | Bad words check | BADWORD | administrator |
| | I | 9/18/2008 5:36:13 PM | Bad words check | BADWORD | administrator |
| | I | 9/18/2008 4:28:58 PM | Authentication | AUTHENTICATIONSUCC | administrator |
| | I | 9/18/2008 4:17:25 PM | Import objects - site 'CorporateSite' | IMPORT | public |
| | E | 9/18/2008 4:17:15 PM | Application_End | EXCEPTION | |
| | I | 9/18/2008 4:16:33 PM | Application_Start | INFORMATION | |

## Cultural versions of bad words

You can set for which cultures will a certain bad word be used. If you choose to **Edit** () a bad word in the list in **Administration -> Bad words** and switch to its **Cultures** tab, you will be offered the following two options:

- **The word is not allowed in all cultures** - the bad word will be filtered in all site cultures
- **The word is not allowed only in following cultures** - the bad word will be filtered only in cultures selected by the check-boxes below

Click **Save** for the settings to take effect.

## 27.4 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 for it
- **Replace** - the bad word is removed and replaced with a predefined word
- **Report abuse** - a report will be created in **CMSDesk -> Tools -> Abuse report**
- **Request moderation** - the post will have to be approved; this also happens in case that the forum, message board or blog comments are not moderated by default
- **Deny** - a warning message will be displayed when the user tries to post the inadequate text, telling which words are to 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** | • | • | • | • |

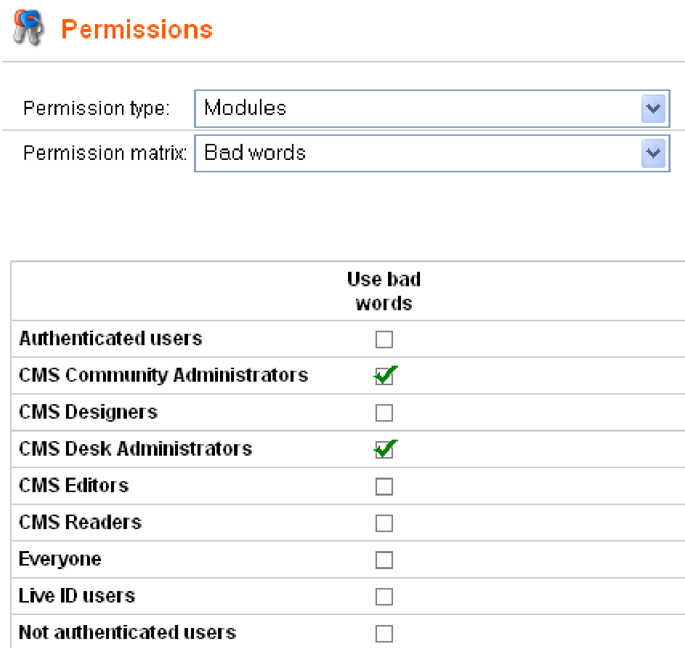
In case that there is **more than one bad word detected** in an entered text and the words have **different actions** set, the actions will be taken according to their hierarchy. Remove, Replace and Report abuse actions have the same "strength", which means that the actions can be taken simultaneously. 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 their 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

## 27.5 Security

To enable the Bad words module, go to **Site Manager -> Administration -> Permissions**, select the **Modules -> Bad words** permission matrix and grant user roles with appropriate permissions.




The following table explains the particular permissions:

| | |
|---|---|
| Use bad words | Users with this permission assigned will not be checked for bad words use, they can use bad words without any limitation. |

# Part XVI

# 28 Module Banned IPs

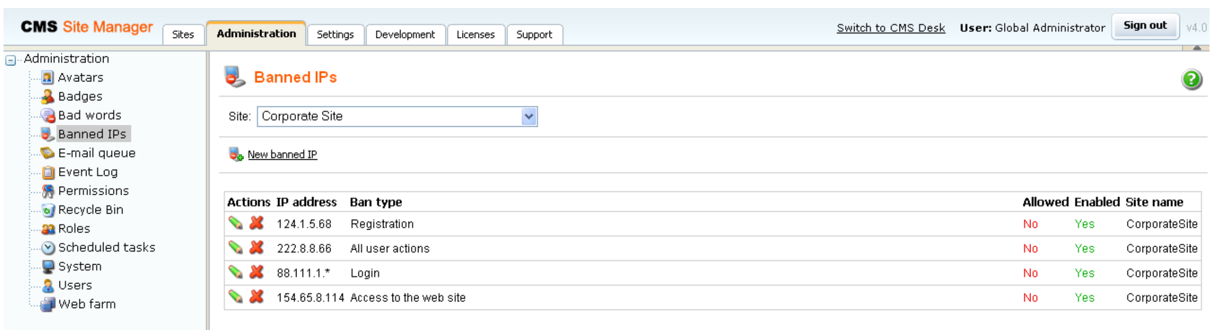
## 28.1 Banned IPs module overview

The Banned IPs module is useful when you want to prevent users from some IP addresses to access the website. This happens typically when the user enters indecent posts into a forum or when she behaves in some other unwanted way. These bans can be set either for separate web sites or globally for all web sites in the system.

## 28.2 Banning an IP address

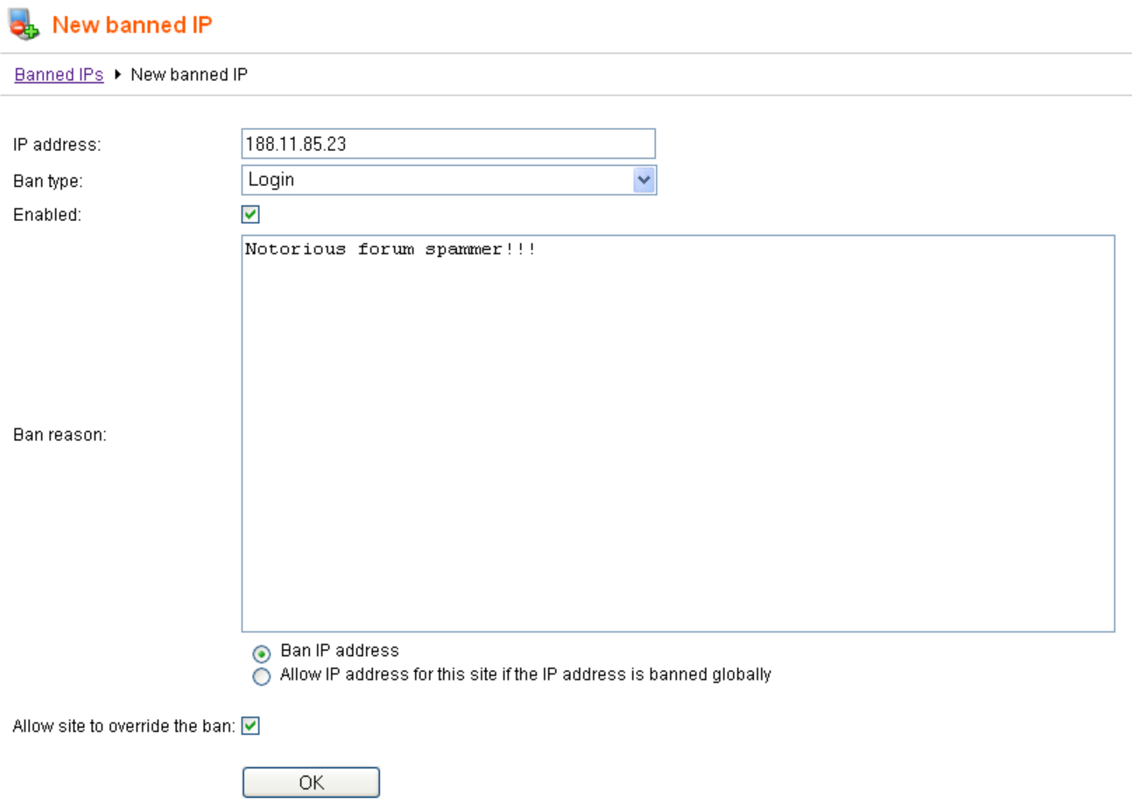
Banned IPs can be managed in **Site Manager -> Administration -> Banned IPs**. Alternatively, you can access the same page from **CMS Desk -> Administration**, but you can manage banned IPs only for the currently edited site.

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.<br>Asterisk ( * ) can be used as a substitute for any number in the IP address, substituting for all values from 0 to 255.<br>Example: 192.168.1.*<br>Dot ( . ) can be used as a substitute for any single number in the IP address, substituting all values from 0 to 9. Example: 192.168.1.15. substitutes all numbers from 150 to 159 |
| Ban type | Type of the ban. The following types are available:<br>• Access to the web site - users cannot access the site from the entered address at all<br>• Login - users cannot log in from the entered address<br>• Registration - users cannot register from the entered address<br>• 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 override | 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.

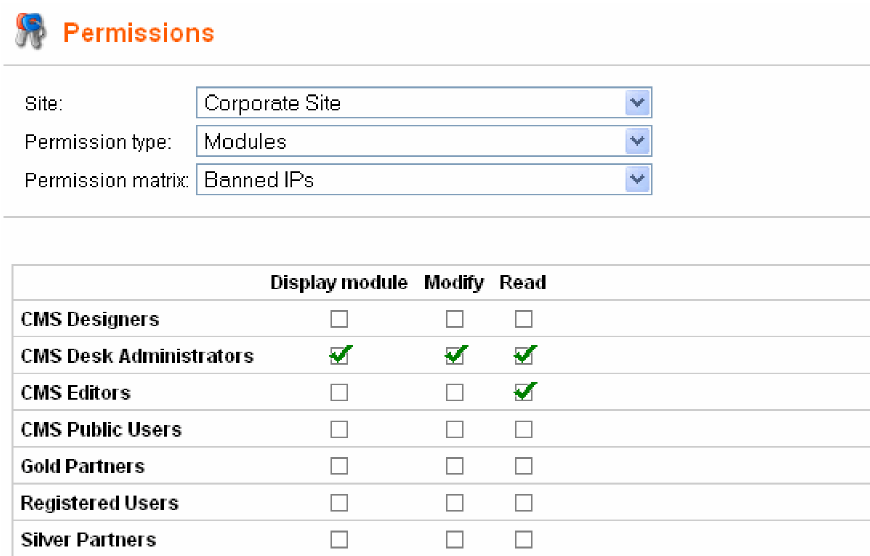
## Editing an existing ban

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**.

## 28.3 Security

To enable the Banned IPs module, you have to make two security settings:

Go to **Site Manager -> Administration -> Permissions**.




The following table explains the particular permissions:

| | |
|---|---|
| Display module | The module is visible in the administration interface. |
| Modify | The user is allowed to create and modify records. |
| Read | The user is allowed to read the records. |

Grant user roles with appropriate permissions according to your needs. After doing so, the module should be displayed and changes could be made to the lists of banned IPs. However, there would still be no effect on the live site.

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 do 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. You can create your own page and set its URL in **Site Manager -> Settings -> Security -> Banned IP redirect URL**.

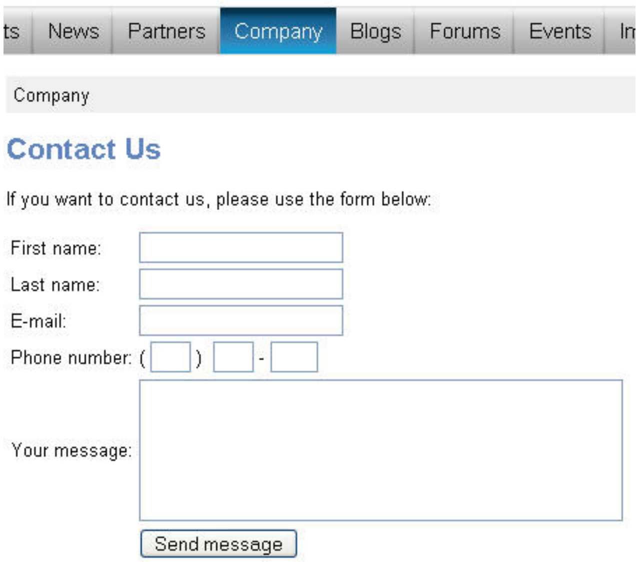
# Part XIX

# 29 Module BizForms

## 29.1 BizForms module overview

The BizForms module enables non-technical users (content editors) to create and publish simple on-line forms, such as "contact us form", "order form" or "event registration form". The form can be published on the web site by inserting a special macro expression into the text or using the BizForm web part.

Here's an example of such form:




The captured data is stored in a separate database table and optionally sent by e-mail to specified address. They can be viewed through the web user interface and exported to Microsoft Excel.

You can find the BizForms module in **CMS Desk -> Tools -> BizForms**.

## 29.2 Creating a new form

In the following chapters, we will create a new form, publish it and read the entered data.

Go to **CMS Desk -> Tools -> BizForms** and click **New BizForm**. Enter the following details:

- **Form display name:** Event registration
- **Form code name:** EventRegistration
- **Table name:** Form_EventRegistration

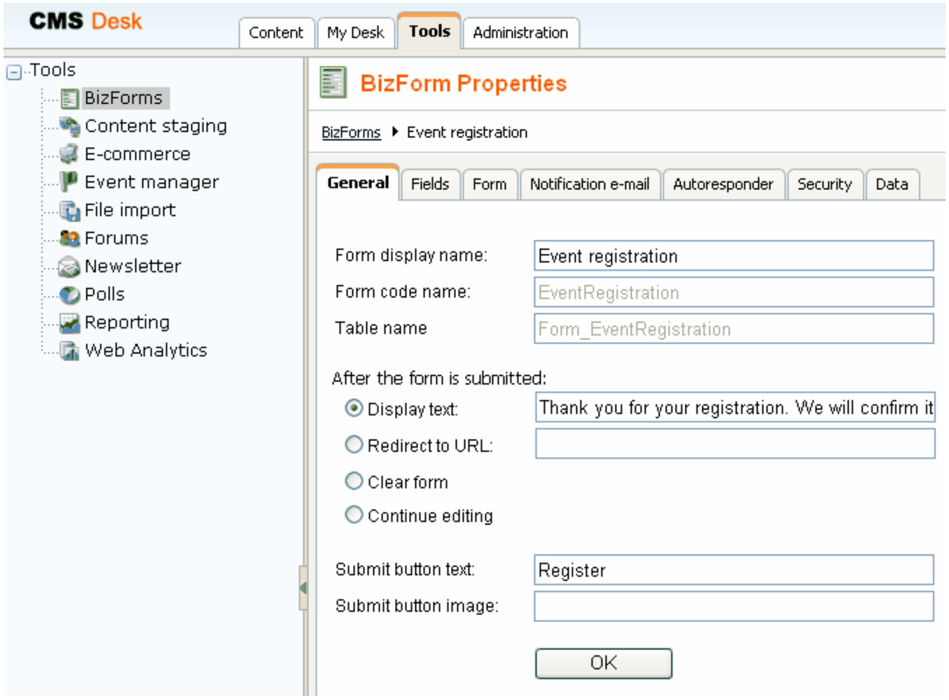
Click **OK**. On the General tab, in the **After the form is submitted** section, choose the **Display text** option and enter the following value:

- **Display text:** Thank you for your registration. We will confirm it shortly by e-mail.

Enter the caption of the submit button:

- **Submit button text:** Register

Click **OK**.




Now we will define the form fields. Go to the **Fields** tab. Add the following fields using the **Simplified** mode:

- Column name: FirstName
- Show on public form: yes
- Field caption: First name
- Field type: TextBox
- Maximum length: 100
- Allow empty value: no

- Column name: LastName
- Show on public form: yes
- Field caption: Last name
- Field type: TextBox
- Maximum length: 100
- Allow empty value: no

- Column name: Phone
- Show on public form: yes
- Field caption: Phone
- Field type: U.S. phone number
- Maximum length: 14
- Allow empty value: **yes**

- Column name: Email
- Show on public form: yes
- Field caption: E-mail
- Field type: E-mail

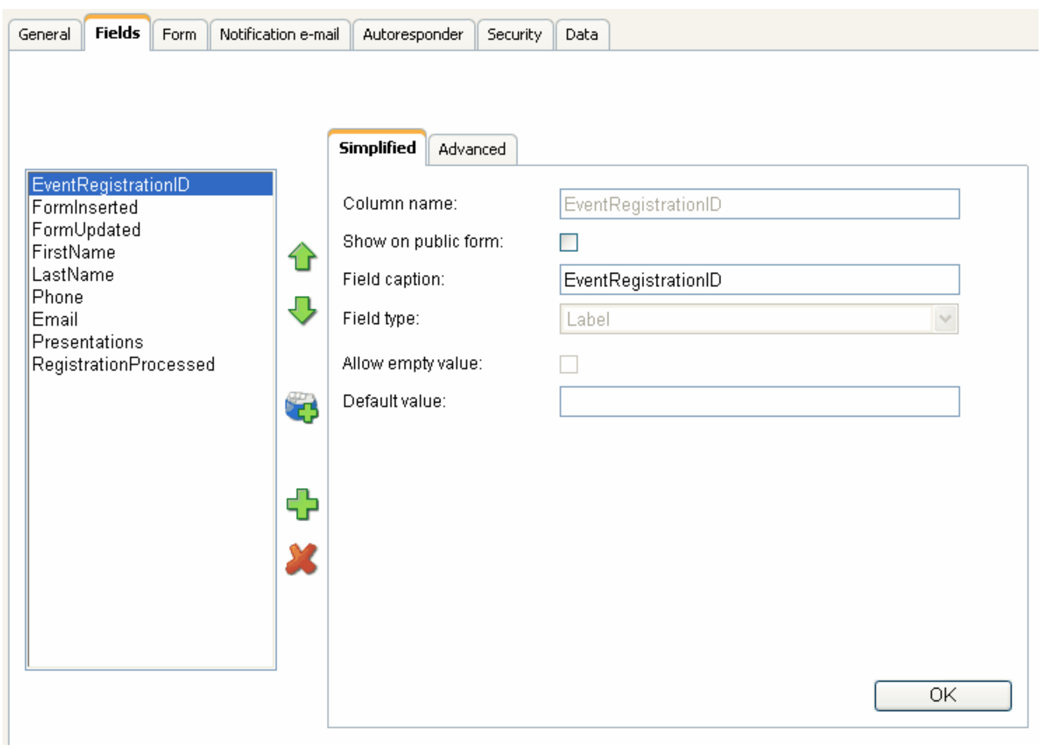
- Maximum length: 100
- Allow empty value: no

- Column name: Presentations
- Show on public form: yes
- 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: **yes**

- Column name: RegistrationProcessed
- Show on public form: **no**
- Field caption: Registration processed
- Field type: Check box
- Allow empty value: **yes**

The last item will be used only by site owners to mark the processed registration forms.

**Please note:** if you switch to the Advanced mode, you can set up additional options, such as validation rules or design.



**Form text localization**

If you need to display the form on a multi-lingual web site, you can localize the field captions and other text strings using an expression like this:

{$myform.fullname$}

or

{$=Hello|de-de=Hallo|it-it=Ciao$}

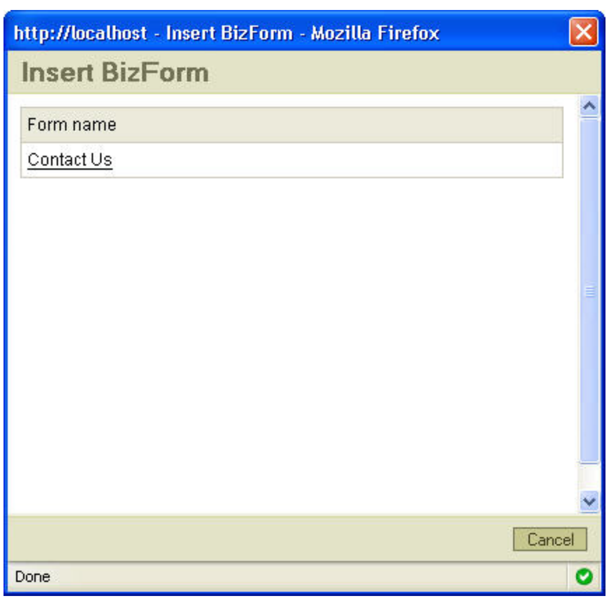
You can find more details in chapter Localization Expressions.

## 29.3 Displaying form on the web site

In order to display the BizForm on the web site, you need to use one of the following options:

- place the BizForm **into WYWIWYG editor** content using the Insert BizForm button
- place the BizForm **into any editable text** using the macro in format **%%control:BizFormControl?BizFormName%%** where BizFormName is the code name of the form
- place the **BizForm web part** and set its Form name property

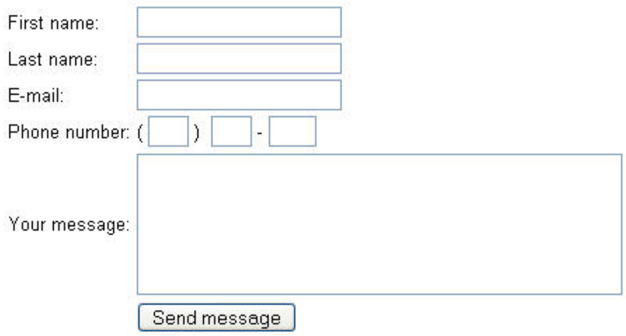
We will use the first option now. Go to **CMS Desk -> Content** and edit some page with editable regions. Click **Insert BizForm** button in the editor toolbar. The Insert BizForm dialog opens. Click **Contact Us** to insert the form into the text:




The following code will be pasted to the region:

%%control:BizFormControl?ContactUs%%

Click **Save** and click **Live site**. You will see the form on the page:




Enter some values and click the submit button to save it.

## 29.4 Sending e-mails

The BizForms module allows you to send to types of e-mails when a new record is added:

- Notification e-mail to the site owner
- Autoresponder - a notification e-mail to the person who submitted the new record in the form

### Notification e-mails

You can configure the notification e-mails on the **Notification e-mail** tab in the form properties.

Check the box **Send form data to e-mail** and enter the following values:

- **From e-mail:** {%Email%} *(this ensures that the e-mail is sent from the person who entered the data and you can easily reply the e-mail)*
- **To e-mail:** webmaster@mydomain.com *(use your e-mail here)*
- **Subject:** Event registration by {%FirstName%} {%LastName%}
- **Attach uploaded documents:** Yes
- **Use custom form layout:** Yes

Click **Generate table layout** and then click **Save**.

When you submit the form now, you will receive a notification with entered data.

### Autoresponder

You can configure the automatic reply on the Autoresponder tab in the form properties. First, you need to choose the field that contains the e-mail address where the automatic response should be sent (typically the E-mail field where the visitor enters his/her e-mail).

Then, you need to enter the sender e-mail and subject and define the body of the e-mail by inserting text and special macros into the text. For example:

*Thank you for your registration. Your registration number is $$value:EventRegistrationID$$ and we will use e-mail address $$value:Email$$ for further communication.*

The macro `$$value:<fieldname>$$` inserts value of the given field into the text.

You can also add attachments that are sent with message, such as event agenda, white papers, etc.

**Inserting dynamic values into e-mail messages**

You can make the values of the following fields dynamic:

- From e-mail
- To e-mail
- Subject
- Display text
- Redirect to URL

You only need to use special expressions in format {%fieldname%} into the field value. When the form is submitted, the value is automatically merged with current form data.

**Example:**

If you enter the **Display text** value like this:

*Dear {%FirstName%}, thank you for your message. We will contact you shortly.*

It will be displayed like this for **First name** "Jane":

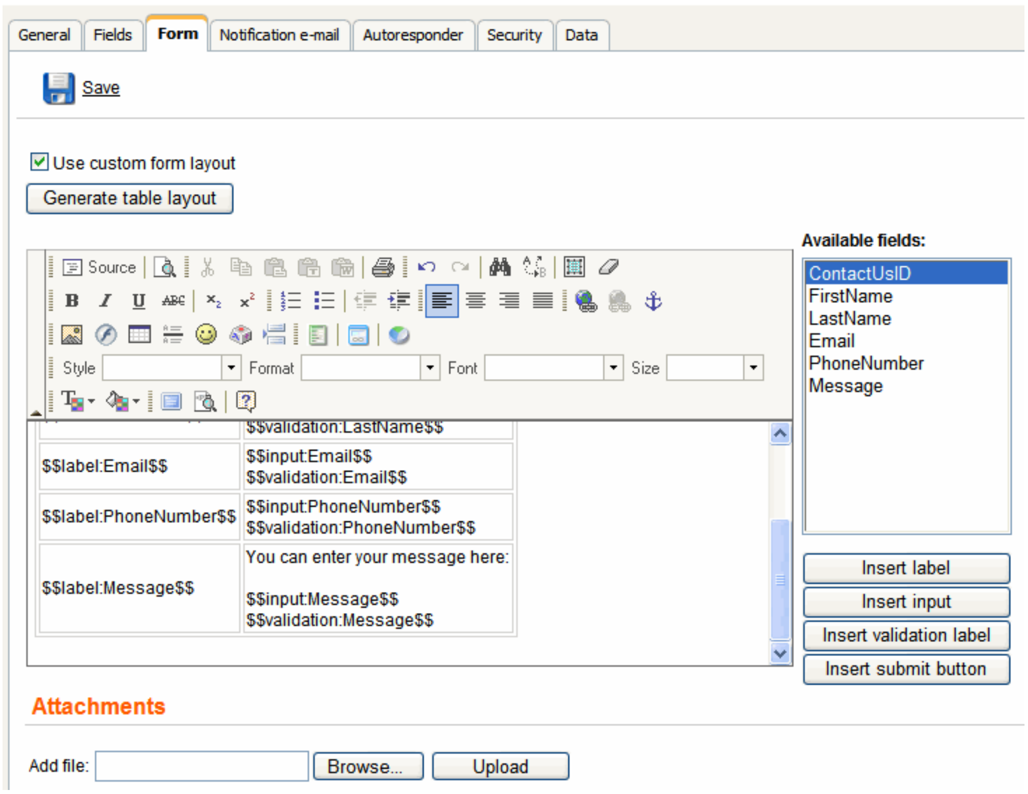
*Dear Jane, thank you for your message. We will contact you shortly.*

It's also useful if you need to **customize the sender address of the notification e-mail**. When the site owner replies to such e-mail, the reply will be sent to the person who submitted the form instead of sending it to some system e-mail address. All you need to do is to set the **From e-mail** value to e.g. **{%Email%}**.

## 29.5 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. Go to **CMS Desk -> Tools -> BizForms** and click the **Edit** icon next to the **Contact us** form. Switch to the **Form** tab and check the **User custom form layout** box. Click **Generate table layout** to quickly generate the fields into the form. Alternatively, you can click the **Insert label/input/validation label** buttons on the right.

You can for example add the text "You can enter your message here:" at the top of the **Message** field:




Click **OK** to save the form.

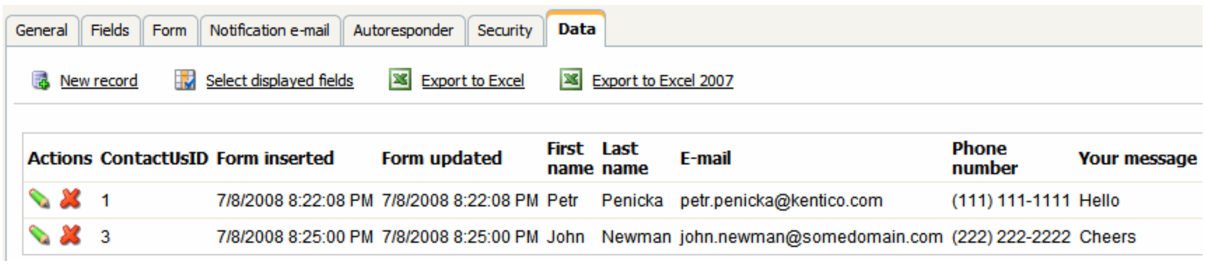
Now the form looks like this:

### Adding Images to the form layout

You can upload an image using the dialog below the editor and then paste the image to the form.

## 29.6 Managing data

The entered data can be managed on the **Data** tab:

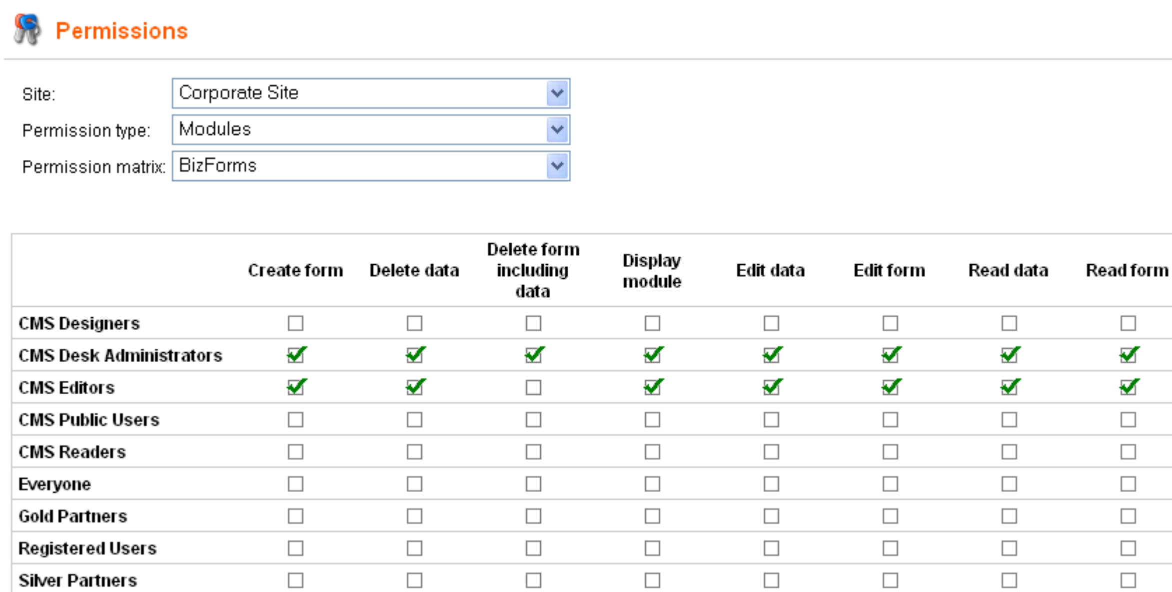



| Actions | ContactUsID | Form inserted | Form updated | First name | Last name | E-mail | Phone number | Your message |
|---|---|---|---|---|---|---|---|---|
| | 1 | 7/8/2008 8:22:08 PM | 7/8/2008 8:22:08 PM | Petr | Penicka | petr.penicka@kentico.com | (111) 111-1111 | Hello |
| | 3 | 7/8/2008 8:25:00 PM | 7/8/2008 8:25:00 PM | John | Newman | john.newman@somedomain.com | (222) 222-2222 | Cheers |

Here you can add, edit and delete records. You can also export data into Excel.

## 29.7 Security

The access to the BizForms module can be managed in **CMS Desk -> Administration -> Permissions**, after you select **Modules -> BizForms** permission matrix:



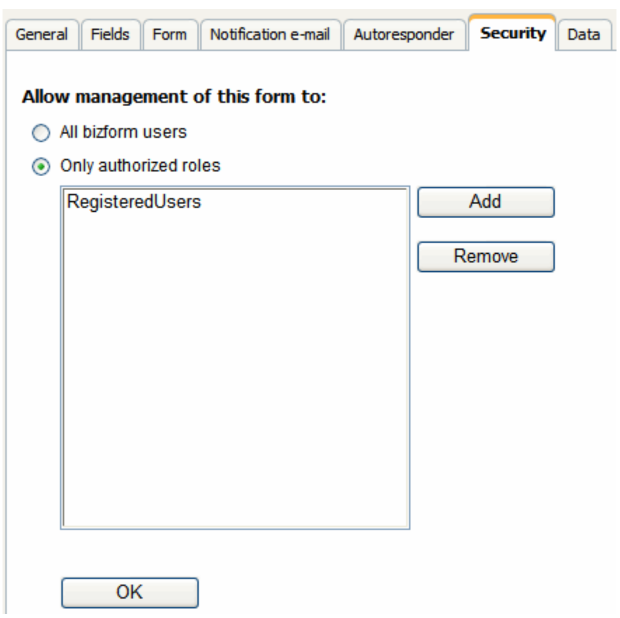

| | Create form | Delete data | Delete form including data | Display module | Edit data | Edit form | Read data | Read form |
|---|---|---|---|---|---|---|---|---|
| CMS Designers | | | | | | | | |
| CMS Desk Administrators | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CMS Editors | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| CMS Public Users | | | | | | | | |
| CMS Readers | | | | | | | | |
| Everyone | | | | | | | | |
| Gold Partners | | | | | | | | |
| Registered Users | | | | | | | | |
| Silver Partners | | | | | | | | |

The following table explains particular permissions:

| | |
|---|---|
| Create form | The user is allowed to create a new form. |
| Delete data | The user is allowed to delete entered records. |
| Delete form including data | The user is allowed to delete the form including data. |
| Display module | The module is displayed to the users in the user interface. |
| Edit data | The user is allowed to create and edit records. |
| Edit form | The user is allowed to edit form configuration, fields and layout. |
| Read data | The user is allowed to read entered records. |
| Read form | The user is allowed to read form configuration, field settings and layout. |

## Security for particular forms

You can specify which roles are authorized to read and modify a form and its data on a form level. Open the properties of the given form and switch to the Security tab. Here you can specify which roles are allowed to use this form:




The form will be hidden to other roles then.

**Please note:** the role must be granted with permissions for the BizForms module. Then, you can further customize the access to particular BizForms. The fact that the role is granted with access to the given form is not sufficient.

# 29.8 BizForms API and internals

## Database structure

Each BizForm has its own database table. You can find its name in the BizForm Properties dialog. This table contains all fields that you specified. You can modify the data using direct access to the database table (there are no dependencies).

## 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 (Site Manager -> Development -> Form controls) and choose the default data type.

## Event Handling

You can run custom actions when a BizForm form is submitted and a record is created, updated or deleted. There are two ways how you can handle the data changes:

**1) Handling the BizForm control events**

In this case, you place the BizForm control on a user control or web part and specify the **FormName** property to the code name of the  required BizForm. 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 properties **BizForm.BasicForm.FieldControls**, **BizForm.BasicForm.FieldLabels** and **BizForm.BasicForm.FieldErrorLabels**.

**2) Handling global data events**

You can write a custom data handler as described in chapter Data handler (CustomDataHandler class). 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 the Form code name in the BizForm Properties dialog).

## Managing data using API

The following examples show you how to create, updated and delete form data using .NET code:

### Creating a new record

**[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);
            }
        }
```

### Updating a record

**[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.";
                }
            }
        }
```

### Deleting a record

**[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.";
                }
            }
        }
```

## 29.9 Adding custom actions to a BizForm form

If you need to add a custom action that will be executed when the form is validated, inserted or updated, you need to create your own clone of the BizForm web part that will contain custom code. You can find an example in chapter Modifying the code or design of standard web parts.

# Part XXX

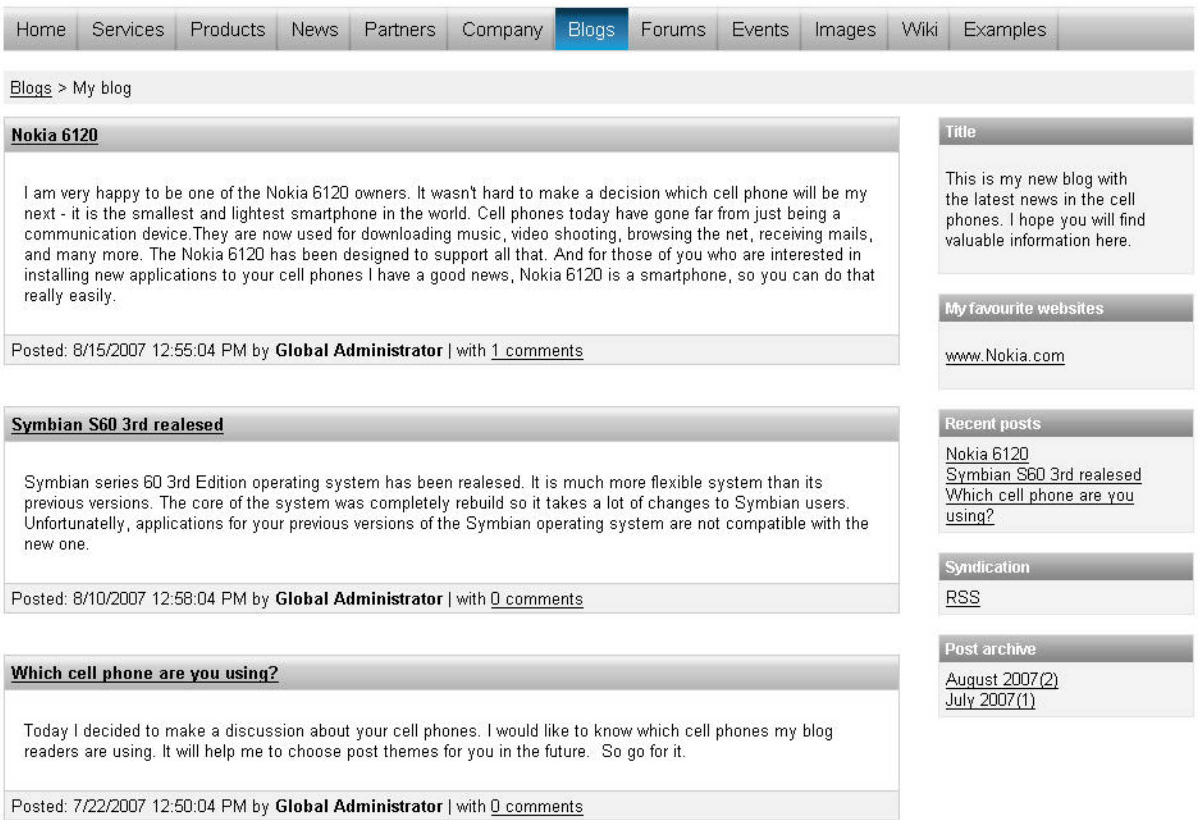
# 30 Module Blogs

## 30.1 Blogs overview

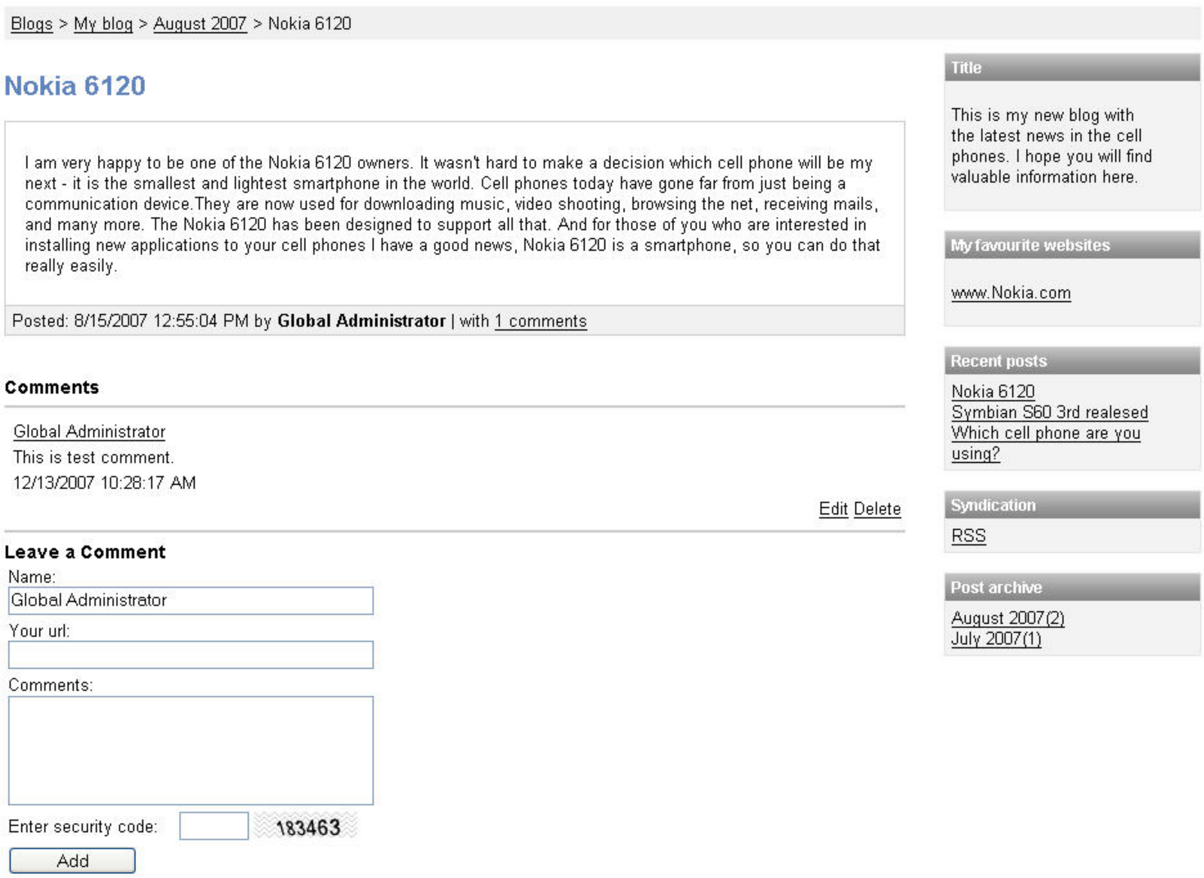
The Blogs module allows you to publish a personal or company blog. You can publish multiple blogs on the same site and there can be multiple editors for each blog.

The blog module fully leverages the standard content management engine, so every blog post you create is a standard document that can be displayed on the web site, searched, etc. You can also configure permissions and workflow for every blog as you're used to do with other content.

The following figure shows the sample blog in the Corporate Site sample web site:



As you can see, the blog displays the list of blog posts and some additional information on the right, such as blog description, RSS feed link, recent posts, favorite web sites and archive. When you click some particular blog post, you can see the text plus the Comments section that allows you to read and add comments to the blog post:




## 30.2 Adding a blog to your site

In this sample, you will learn how to create a new blog in the sample Corporate Site, under the News section.

Sign in to **CMS Desk**. In the **Content** section, click **/News** and click **New**. Choose to create a new **Blog**. Enter the following details:

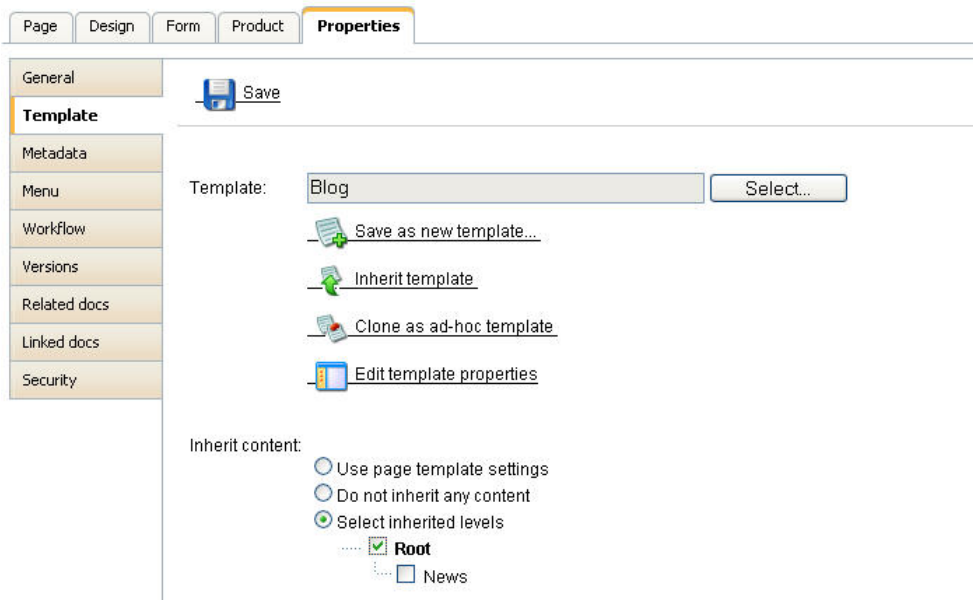
- **Blog name:** My new blog
- **Blog description:** This is my new blog on web design.
- **Side column text:** I like these web sites...
- **Moderate comments:** yes
- **Open comments for:** Always
- **Send comments to e-mail:** *enter your e-mail address*
- **Allow anonymous comments:** yes
- **Use CAPTCHA for comments:** yes (this will require a verification of the number displayed in the picture to avoid spam posts)

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
- **Moderate comments:** indicates if blog comments are moderated
- **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:

**1)** Go to the **Properties -> Template** dialog and click **Template properties**. Set the value **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.



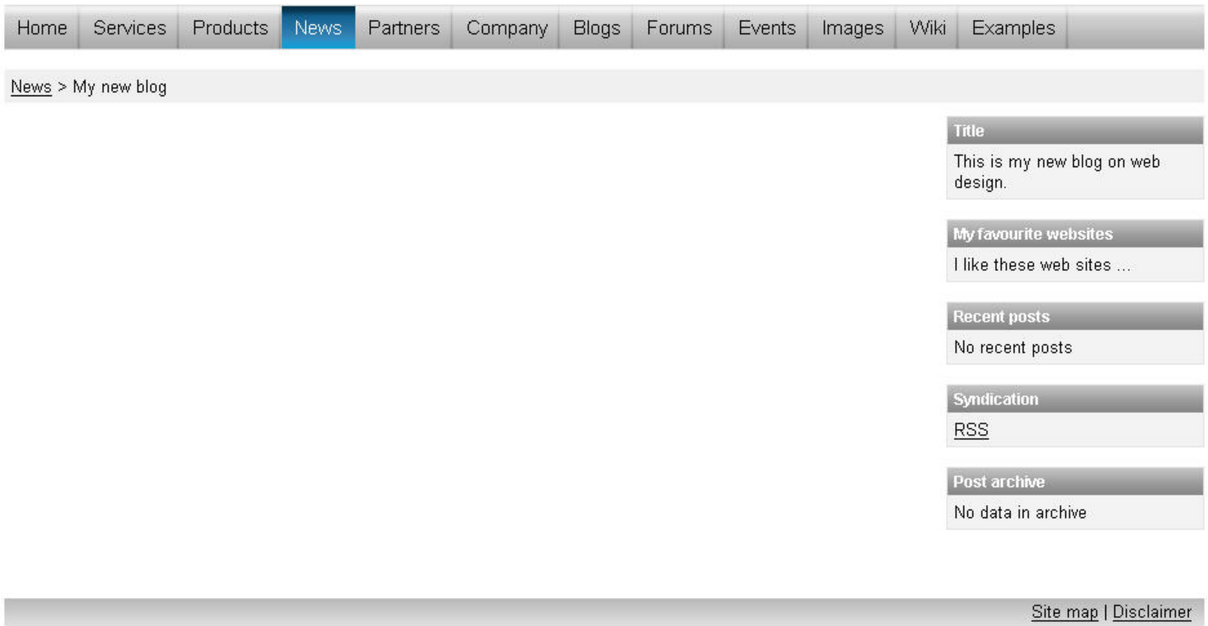

Click **OK**.

**2)** Choose the **root** in the content tree and switch to the **Design** mode. Configure the **menuMain** web part and set the **Document types** value to this:

**`cms.menuitem;cms.blog`**

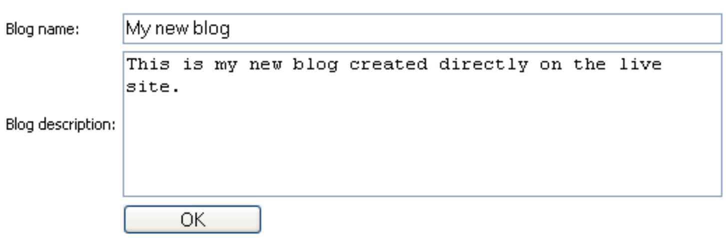
Click OK. It ensures that the main menu contains a link to your blog.

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.



## 30.3 Adding posts to your blog

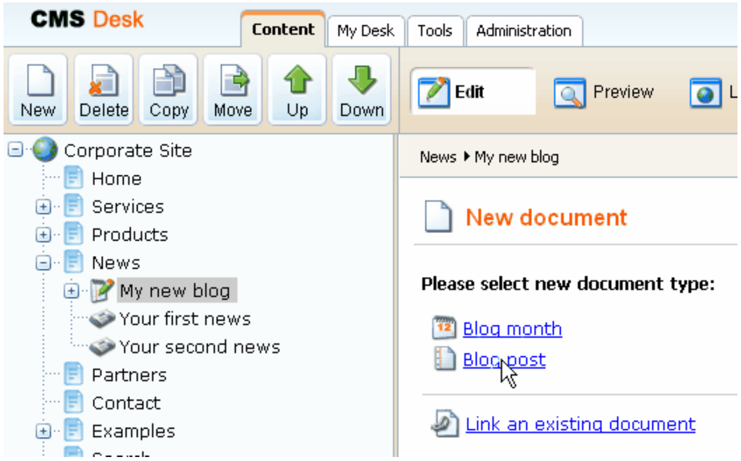
You can add new posts to your blog in two ways:

**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**, locate the blog main document and then click **New**. Choose to create a new **Blog post**:

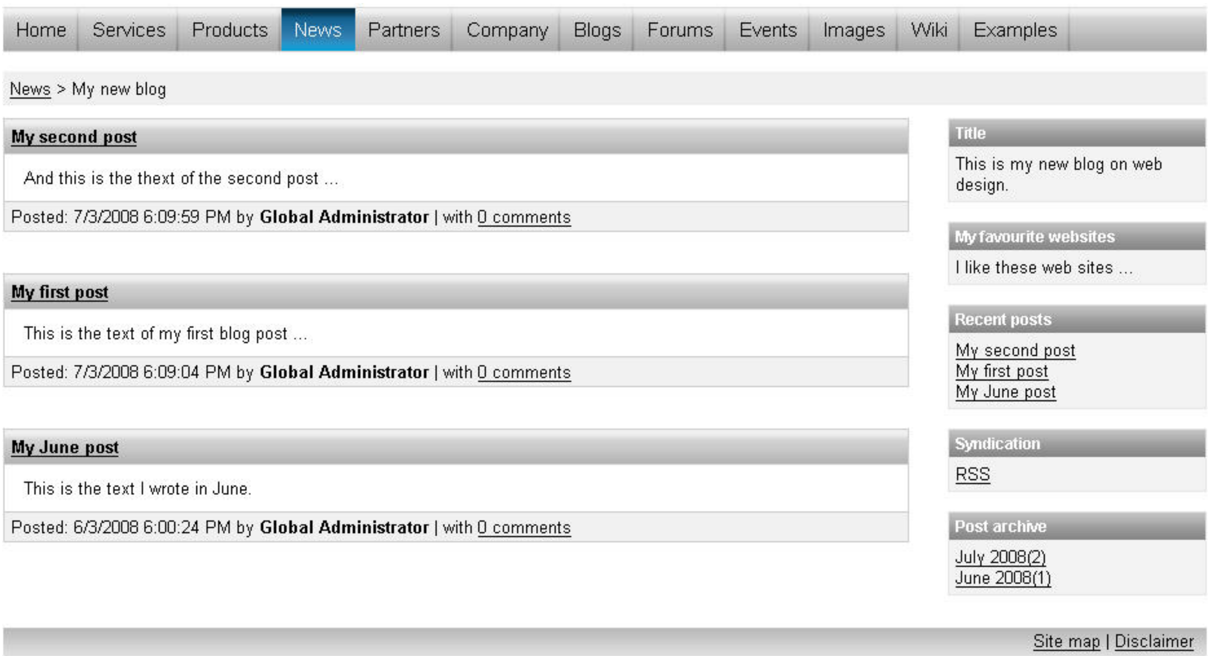



Enter the post content and click **Save**.

## 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:



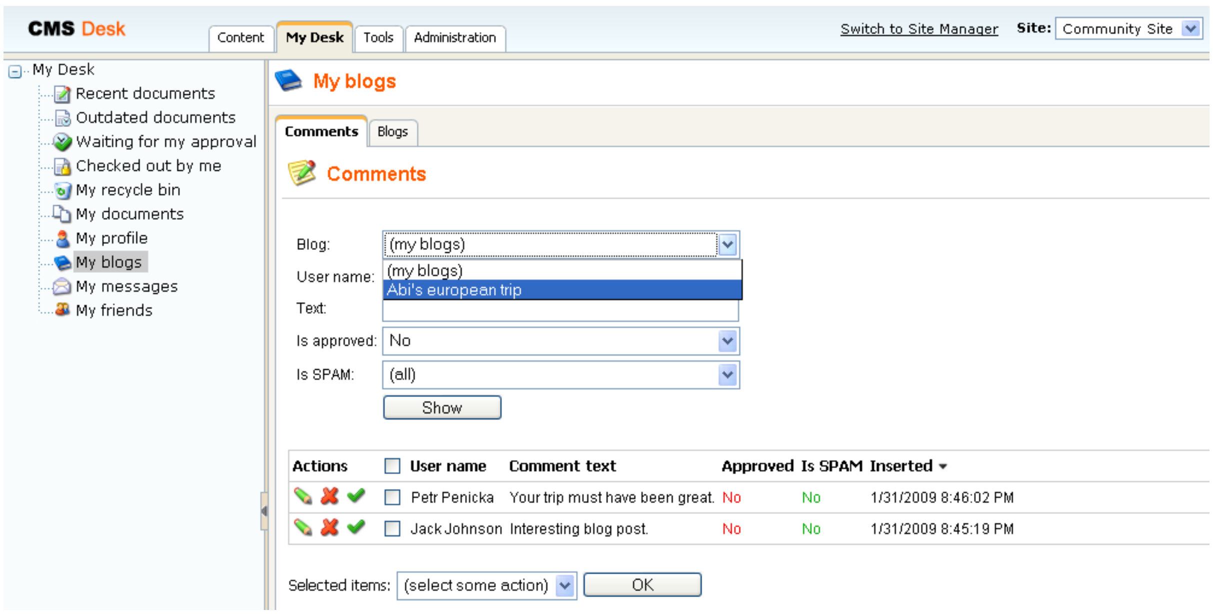
## 30.4 Moderating comments

The comments can be moderated in tree ways:

**1) In the My Desk -> My blogs section**

In this section, the currently logged-in user can manage comments at all her 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**.

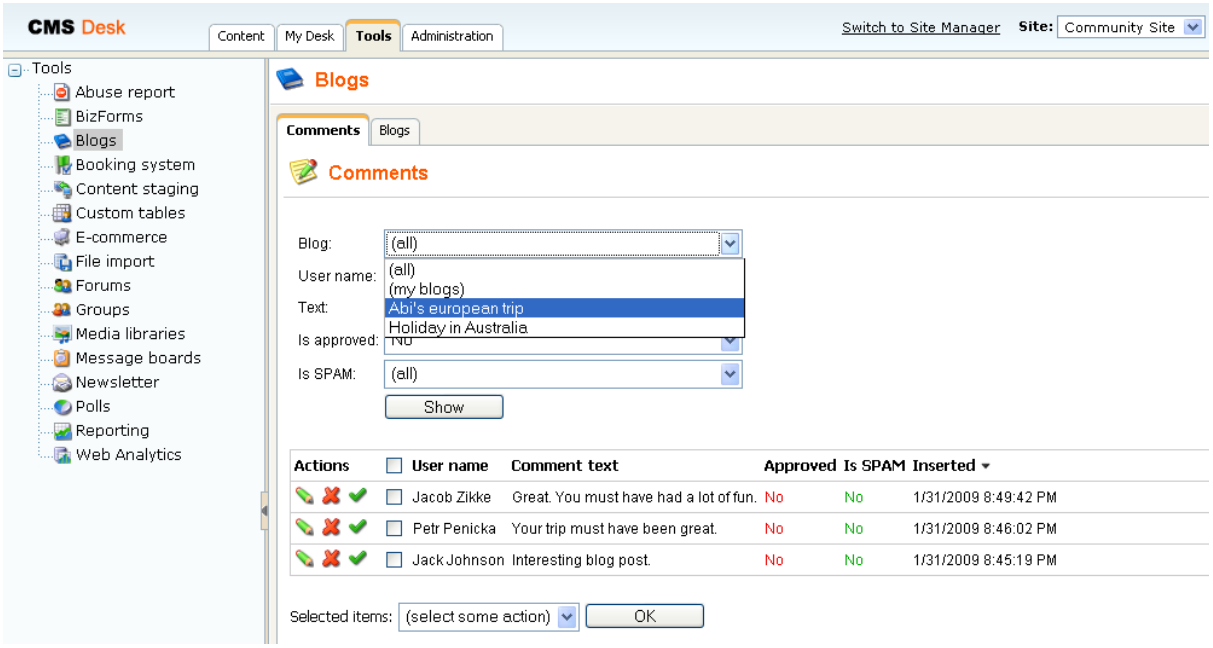



**2) In the CMS Desk -> Tools -> Blogs section**

In this section, management of blog comments on the site is possible based on the permissions:

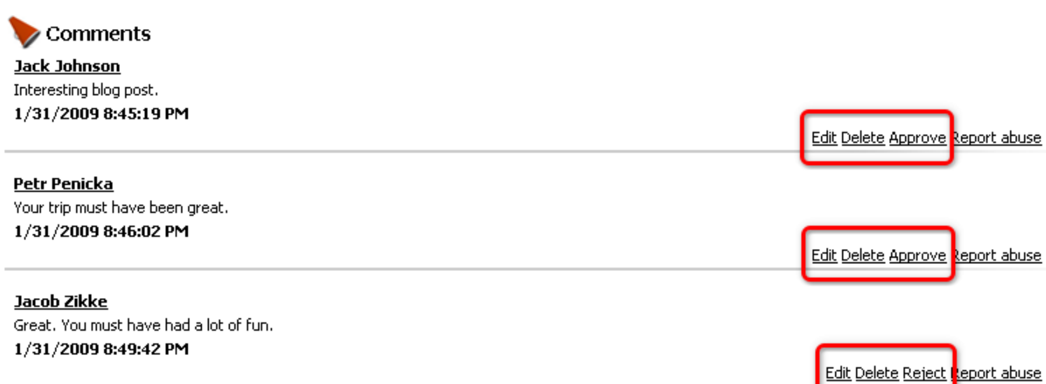
**Global administrators** and users with the **Manage comments** permission for the Blogs module can manage comments of all blogs on the current site. Blog **moderators** and blog **owners** can manage comments of their own blogs and blogs for that they are moderators. These permissions are reflected 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**.



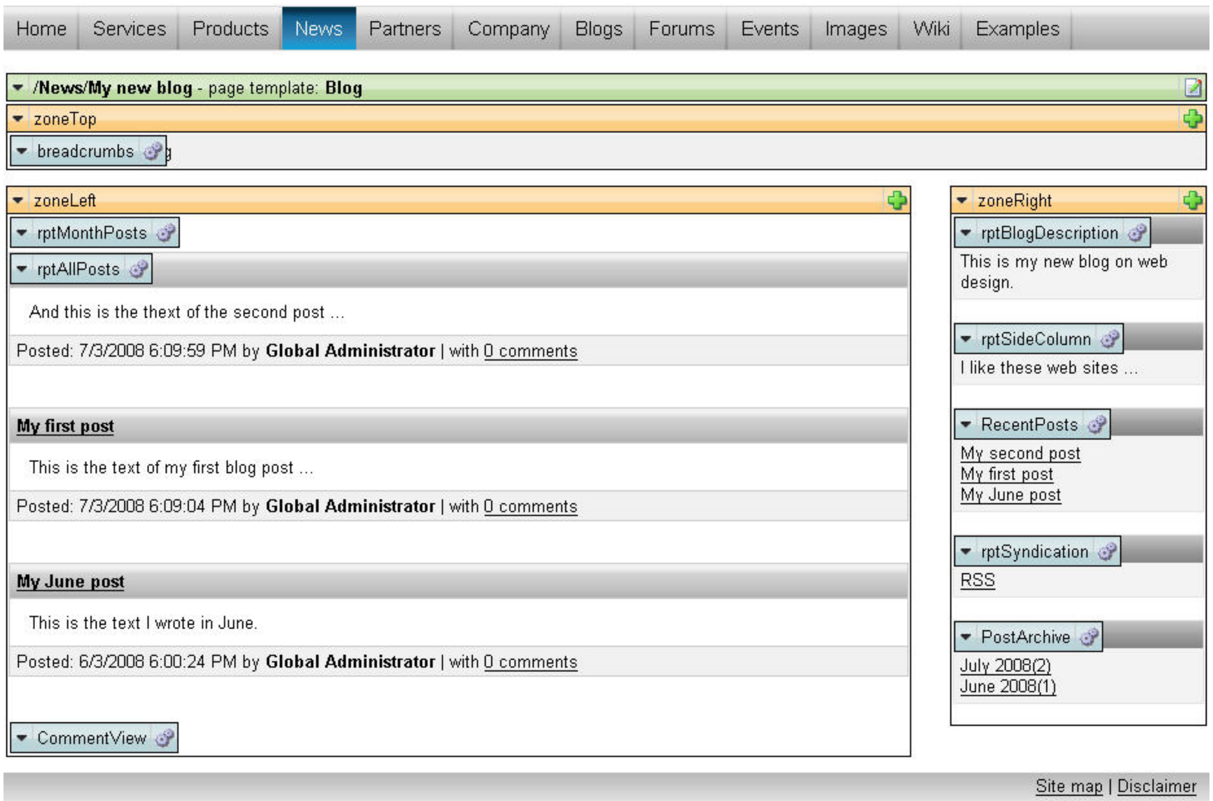

### 3) On-site management

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 owner**, **moderators** of the current blog, **global administrators** and users with the **Manage comments** permissions for the Blogs module.



## 30.5 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

## 30.6 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** mode 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, resp. 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.

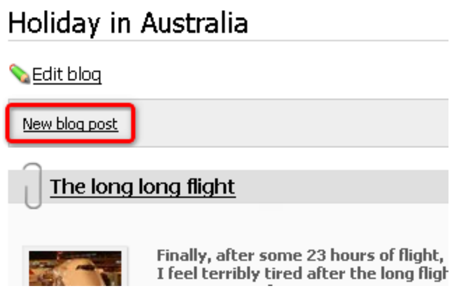

## Add new blog posts

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

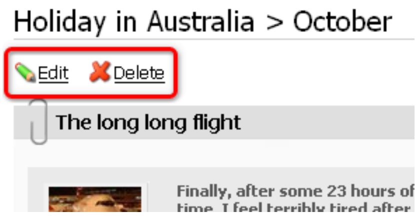
1) Create an **alternative form** for the **Blog post** document type, which will contain the fields that you want to let users modify. If you are not familiar with the Alternative forms concept, please refer to the Alternative forms chapter first.

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.

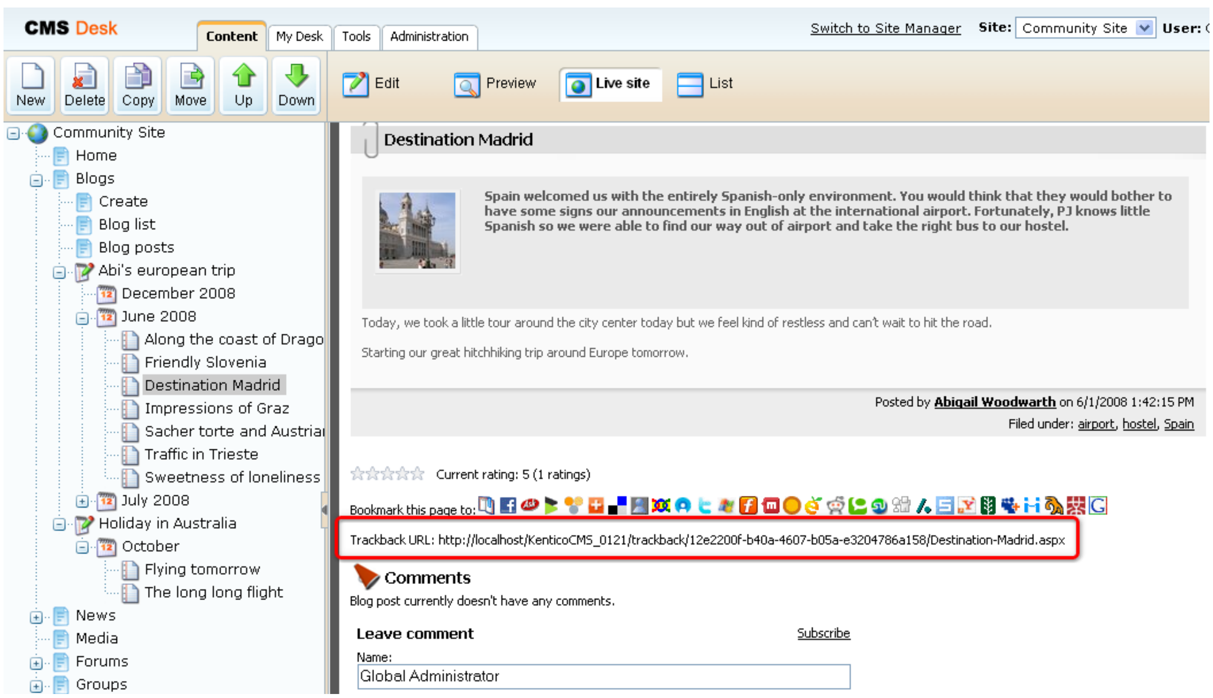


## 30.7 Trackbacks

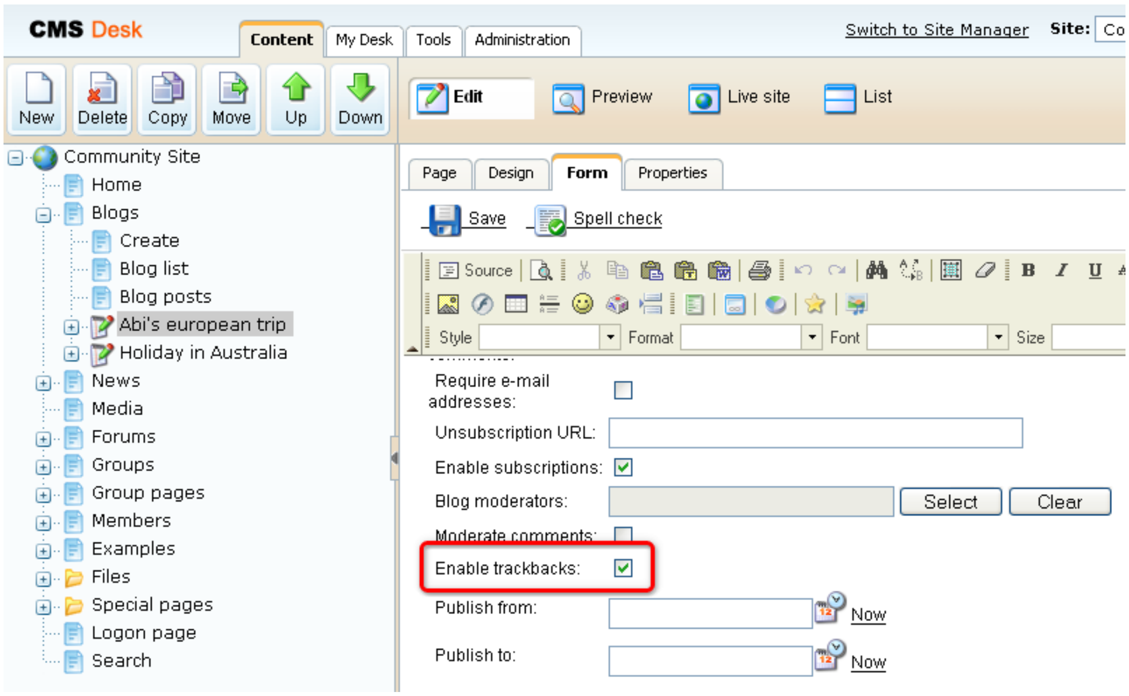
The Trackbacks feature allows sending of blog post links to other blogs (not only within the same site, but also to completely different sites created not only with Kentico CMS) when users create a new blog post. This link is typically, but not always, added as a blog comment, as implemented in Kentico CMS. This is useful when you want to let readers of a topic-related blog post know about your new blog post. You can find detailed information on trackbacks on the following Wikipedia page: http://en.wikipedia.org/wiki/Trackback

### How it works

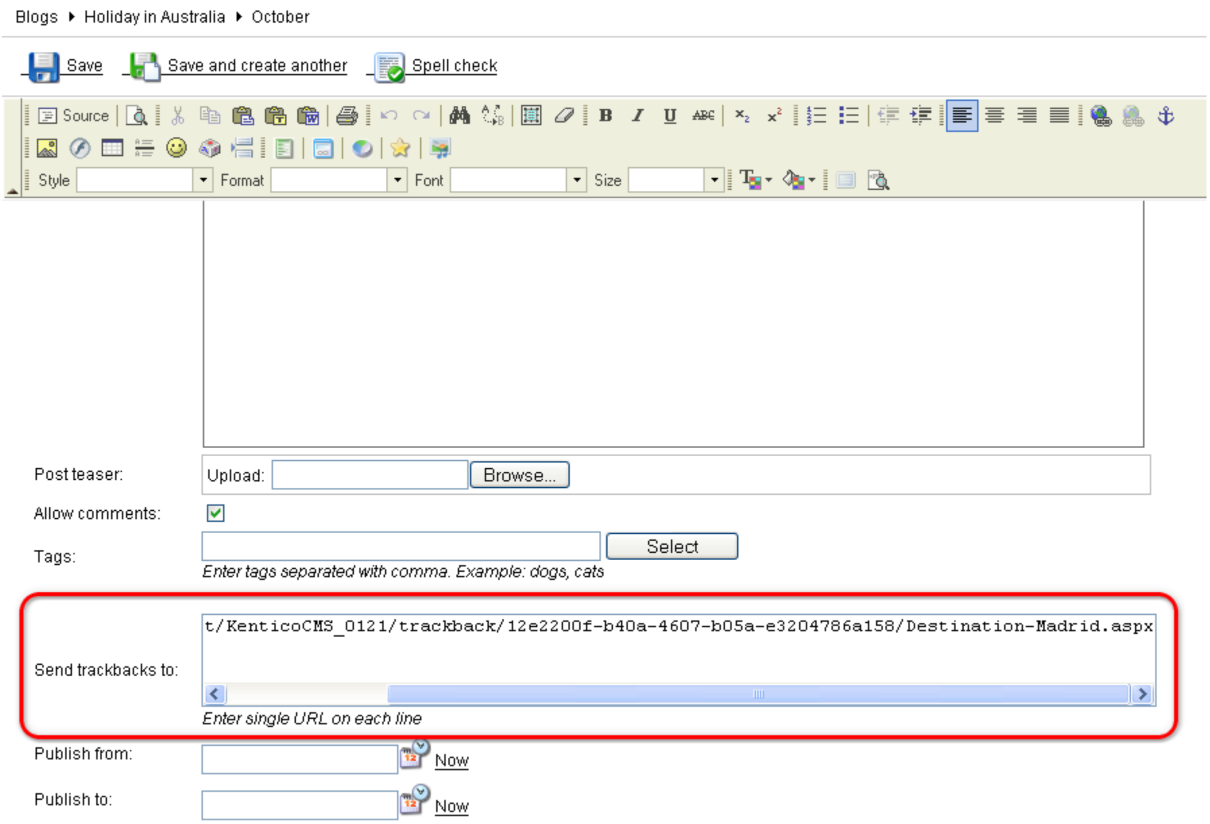
1) There is a Trackback URL published below the blog post. These URLs can be found on various blogs at sites developed not only with Kentico CMS.



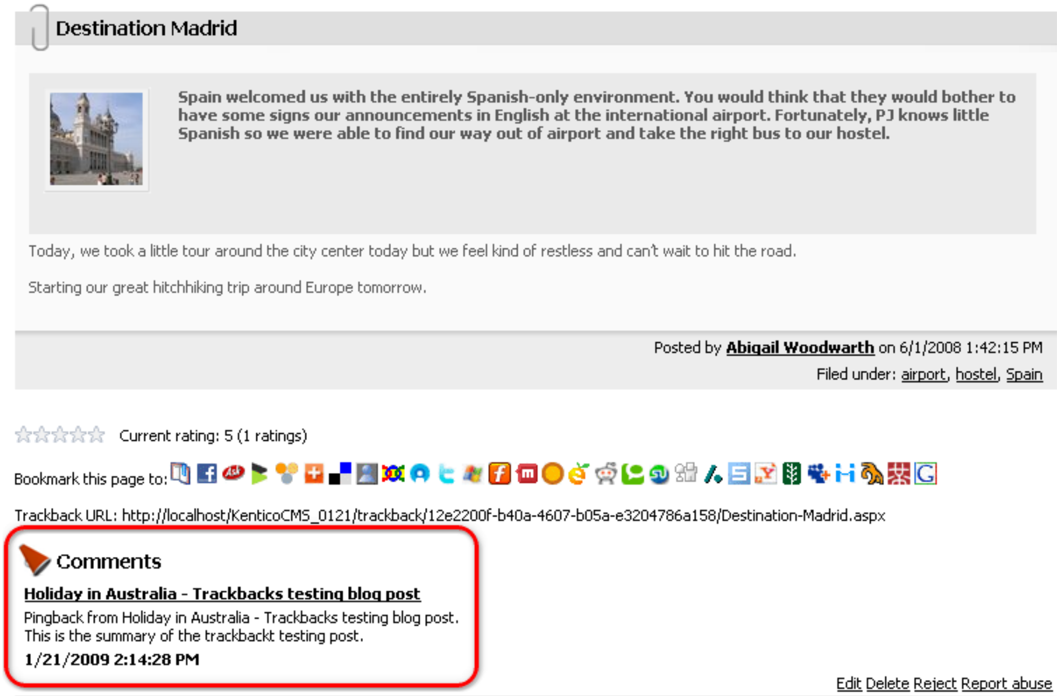
To enable displaying of these links, the **Enable trackbacks** option must be enabled on the blog's **Form** tab. The **Display trackbacks** property of the **Comment view** web part must be enabled too.



2) When users are creating a new blog, you can add such URLs into the **Send trackbacks to** field. More  trackback URLs can be entered, each on a new line.



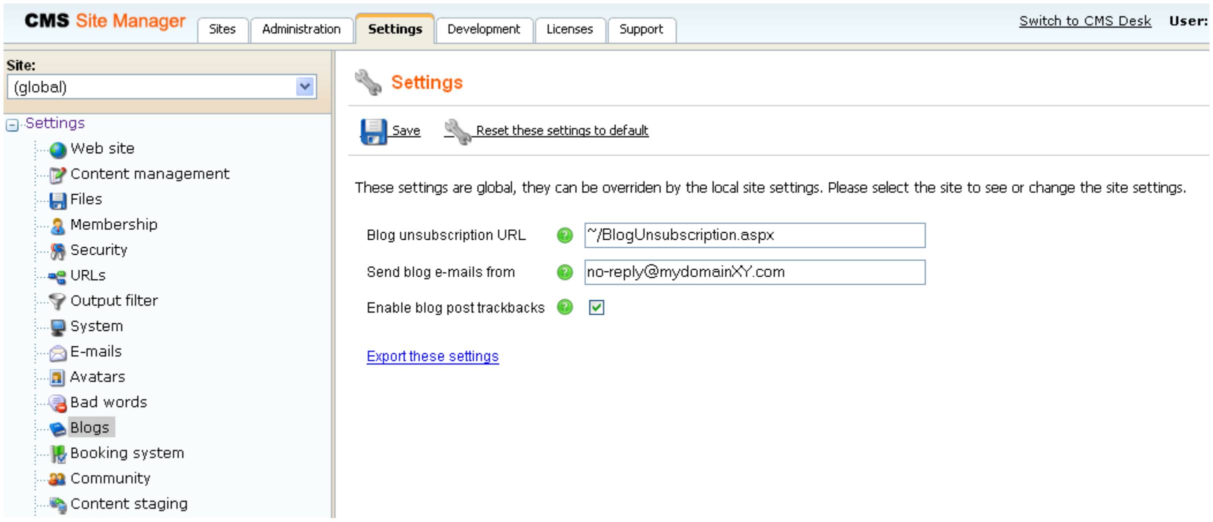
3) When the blog post is submitted, trackback ping is sent to the trackback URLs. If everything is configured correctly at the other blogs, a blog comment will be added with the blog post link, title and summary, as you can see in the screenshot below.



## 30.8 Settings

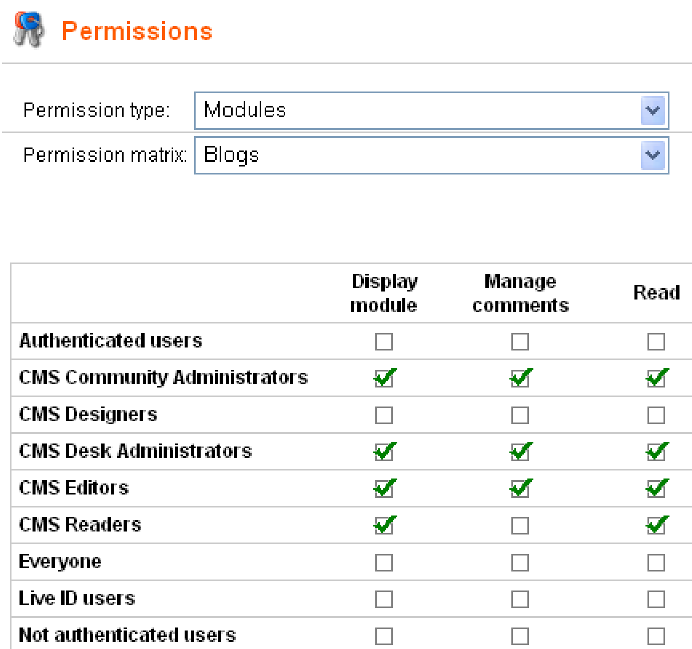
Settings of the Blogs module can be done in **Site Manager -> Settings -> Blogs**. The following settings are available:

- **Blog unsubscription URL** - URL of a page on that 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



## 30.9 Security

The Blogs module leverages the standard security used for all documents. You can find more details in chapter Permissions for modules and documents.




| | Display module | Manage comments | Read |
|---|---|---|---|
| Authenticated users | ☐ | ☐ | ☐ |
| CMS Community Administrators | ☑ | ☑ | ☑ |
| CMS Designers | ☐ | ☐ | ☐ |
| CMS Desk Administrators | ☑ | ☑ | ☑ |
| CMS Editors | ☑ | ☑ | ☑ |
| CMS Readers | ☑ | ☐ | ☑ |
| Everyone | ☐ | ☐ | ☐ |
| Live ID users | ☐ | ☐ | ☐ |
| Not authenticated users | ☐ | ☐ | ☐ |

You can manage the posts if you are either owner of the blog, blog moderator, global administrator or if you are granted with the **Manage comments** for the **Blogs module** in the **Administration -> Permissions** dialog. The **Display module** permission enables the Blogs module in the user interface.

The blogs in the **My Desk -> My blogs** dialog are displayed only to the blog document owner.

## 30.10 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);
               }
            }
        }
```

# 30.11 Blog comments notifications

## 30.11.1 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 are the notification e-mails 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 her e-mail 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 her e-mail 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 her e-mail 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

## 30.11.2 User subscriptions

Users can subscribe to receiving notifications about new blog comments at some blog post. It can be done 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.

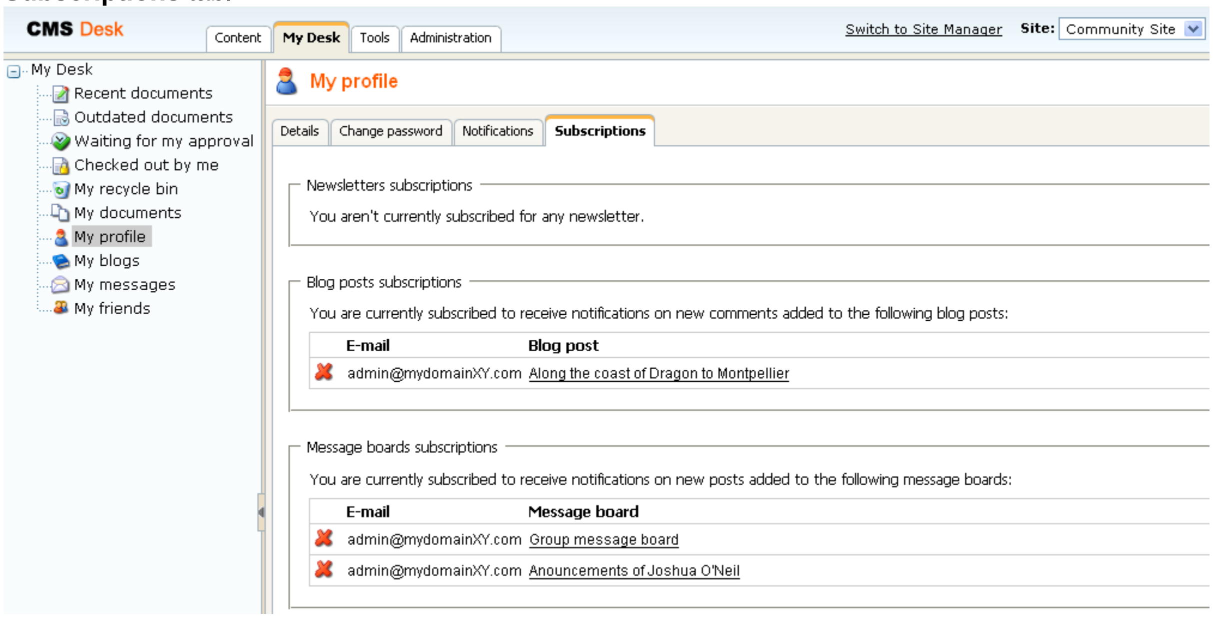

2) Users can also subscribe without leaving a comment, by clicking the **Subscribe** link at the top of the Leave comment form.


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.

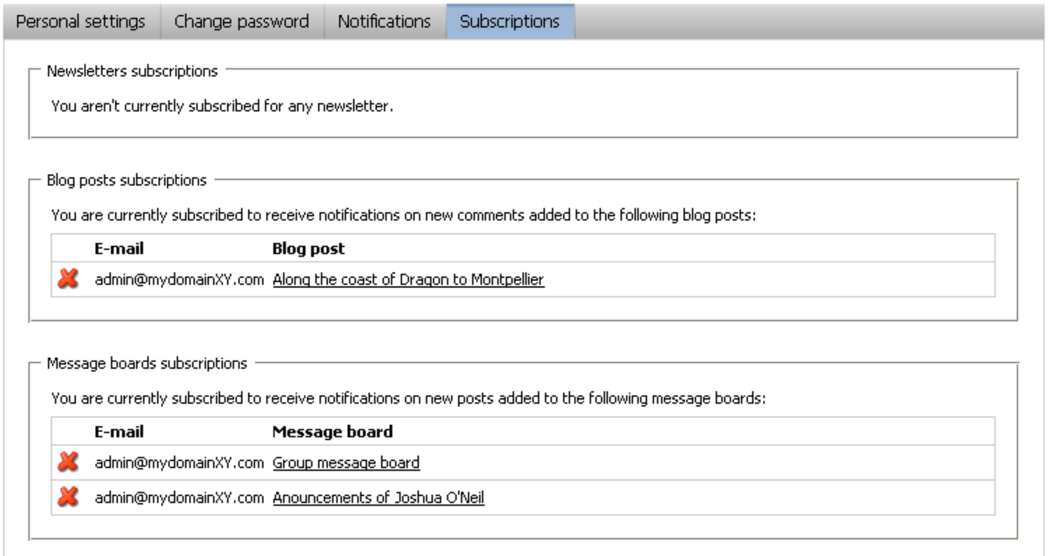
## Subscriptions management

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.

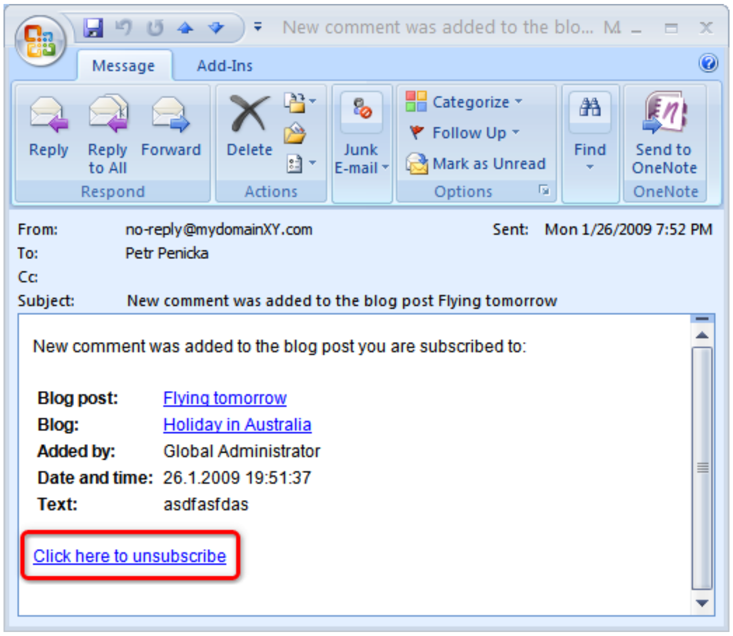


2) On live site, users can view their subscriptions in the **My account** web part. The **Display subscriptions** property of the web part must be enabled for this to be possible.



### 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.

## 30.11.3 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:

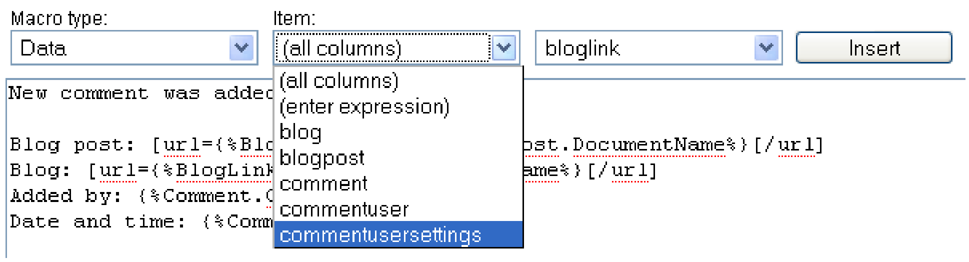
- **Blog owners** - e-mails are based on the **Blogs - Notification to blog owner** e-mail template
- **Blog moderators** - e-mails are based on the **Blogs - Notification to blog moderators** e-mail template
- **Subscribers** - e-mails are based on the **Blogs - Notification to blog post subscribers** 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** - there 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%}

# Part XX

# 31 Module Booking system

## 31.1 Booking system 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.

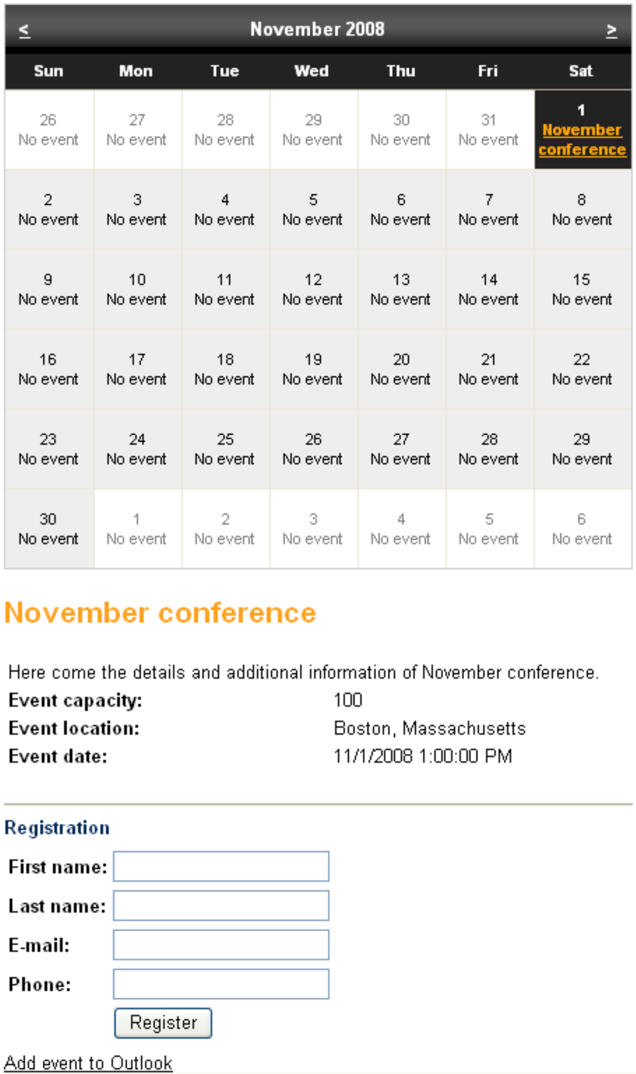
## 31.2 Publishing the events

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 |

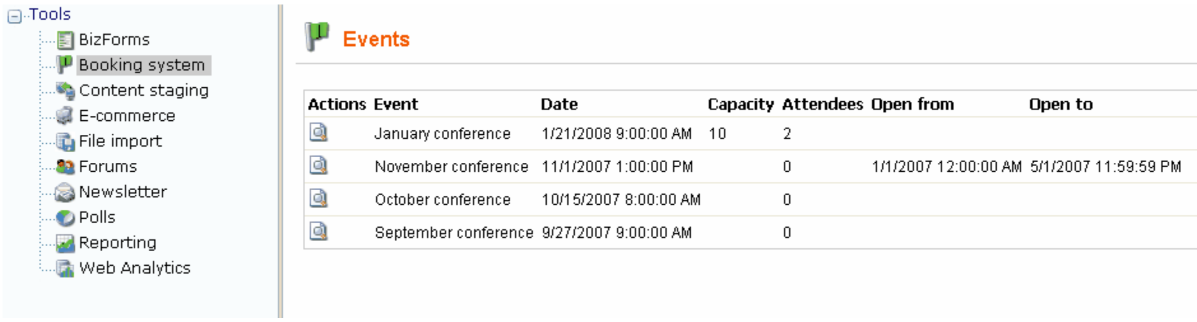
If you wish to publish events, you can use the page template **Events/Event calendar with event details** to display the event calendar and allow users to register or you can build your own page template.

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.



# 31.3 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.

## 31.4 Security

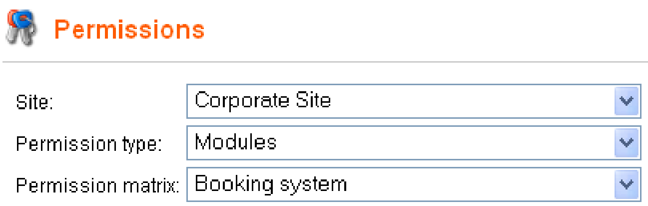
The security is split into two sections:

### 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 and the user need to be granted with permissions for this module in the **Administration -> Permissions** section.




| | Display module | Modify | Read |
|---|---|---|---|
| CMS Designers | ☐ | ☐ | ☐ |
| CMS Desk Administrators | ☑ | ☑ | ☑ |
| CMS Editors | ☐ | ☐ | ☐ |
| CMS Public Users | ☐ | ☐ | ☐ |
| CMS Readers | ☐ | ☐ | ☐ |
| Everyone | ☐ | ☐ | ☐ |
| Gold Partners | ☐ | ☐ | ☐ |
| Registered Users | ☐ | ☐ | ☐ |
| Silver Partners | ☐ | ☐ | ☐ |

You can configure the following permissions:

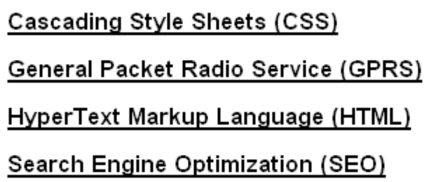
- **Display module -** enables the module in the user interface
- **Modify** - allows users to modify (add, update, delete) the list of attendees and their details
- **Read** - allows users to read the list of attendees, resend messages and send mass e-mail

# Part XX

# 32 Module Categories

## 32.1 Categories module overview

The categories module enables users to categorize their documents. Any document in the content tree can be assigned to an unlimited number of categories. The module contains the Category list web part. This web part can display a list of categories in the form of clickable links. After clicking some of the categories, a list of all documents in the selected category will be displayed.





## 32.2 Creating a new category

There are two types of categories:

- **Global categories** - categories created by the system administrator; can be used by all users
- **Custom categories** - categories created by a user, i.e. her own ones; only the user who created such category can use it

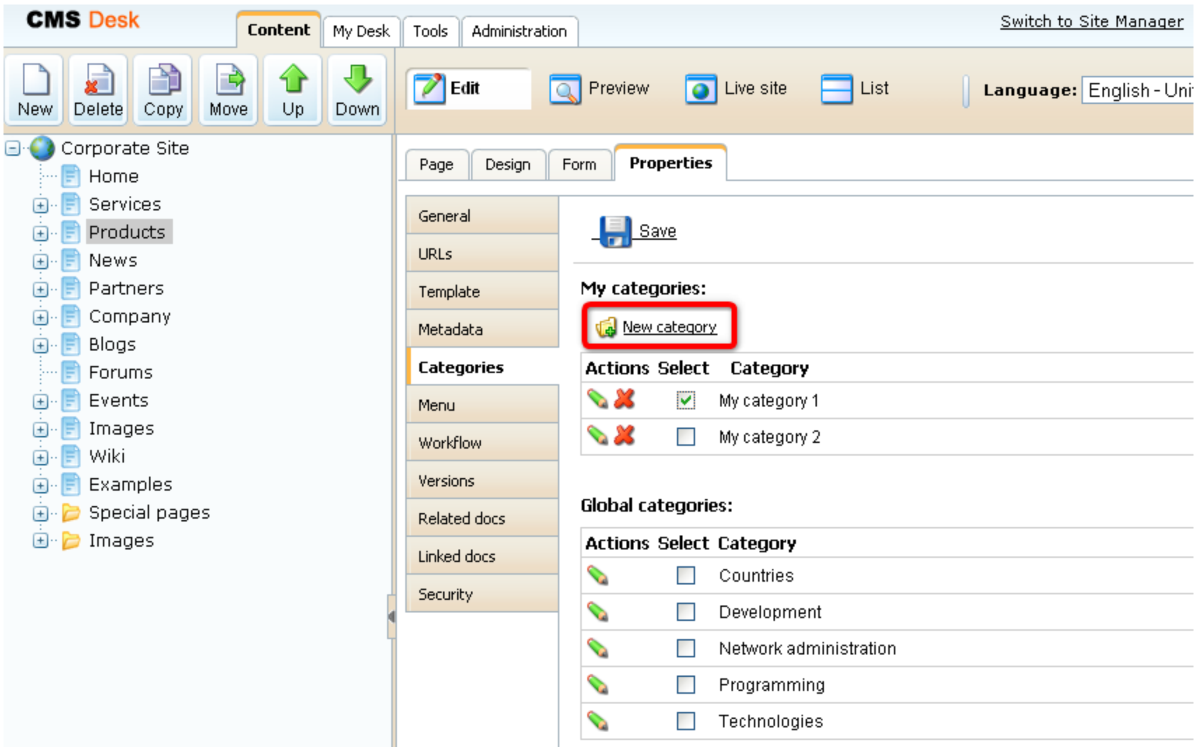
### Custom categories

Each user can create her 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 it's **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:

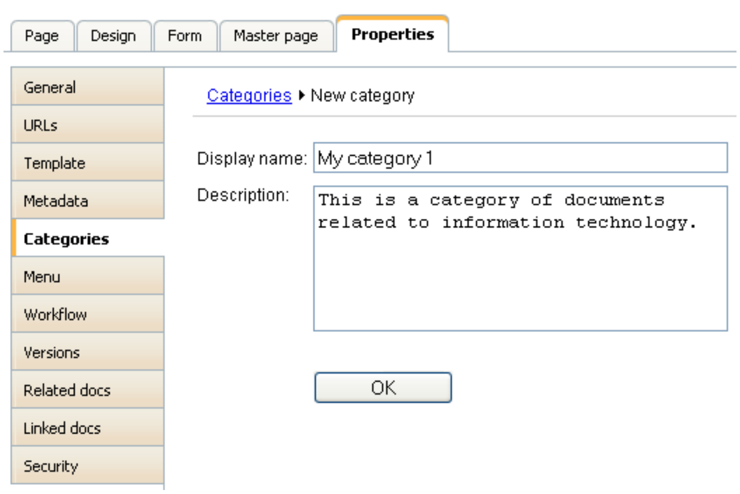
1) Click the **New category** link.



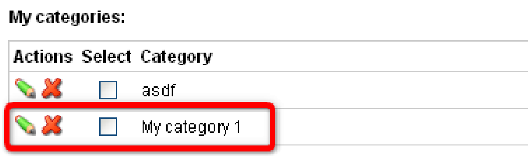
2) Enter the required details:

| | |
|---|---|
| Display name | Display name of the category. |
| Description | Text describing of the category. |

And click **OK**.

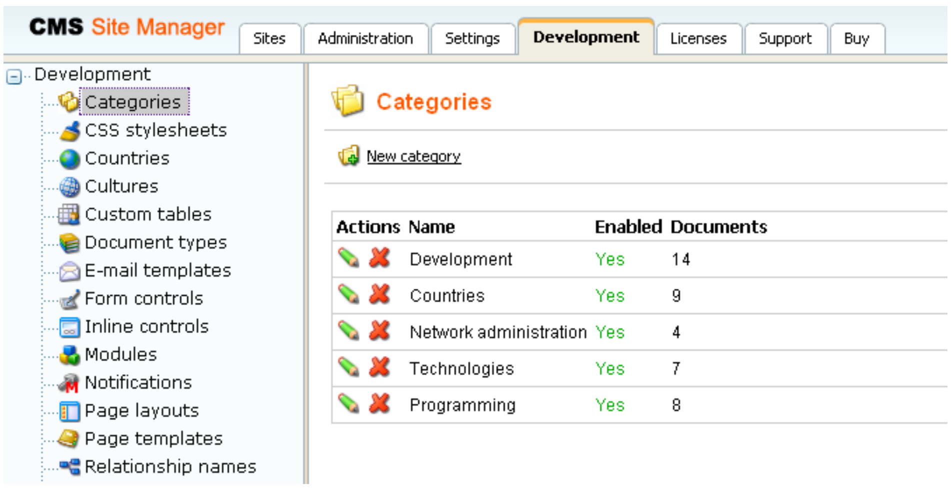


3) You should be redirected back to the list where your new custom category should be visible.



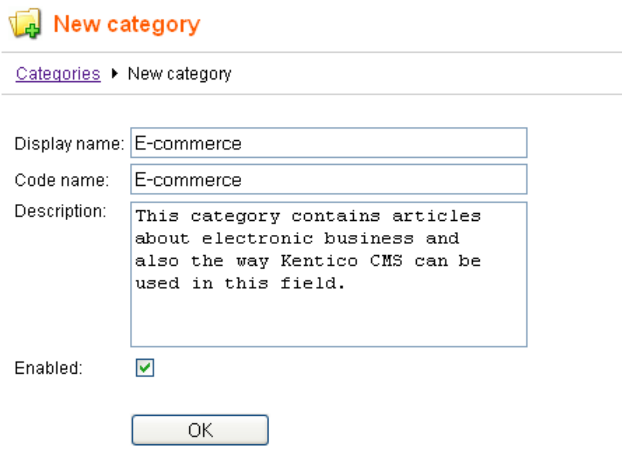

## Global categories

Global categories are created by system administrator 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.



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. |

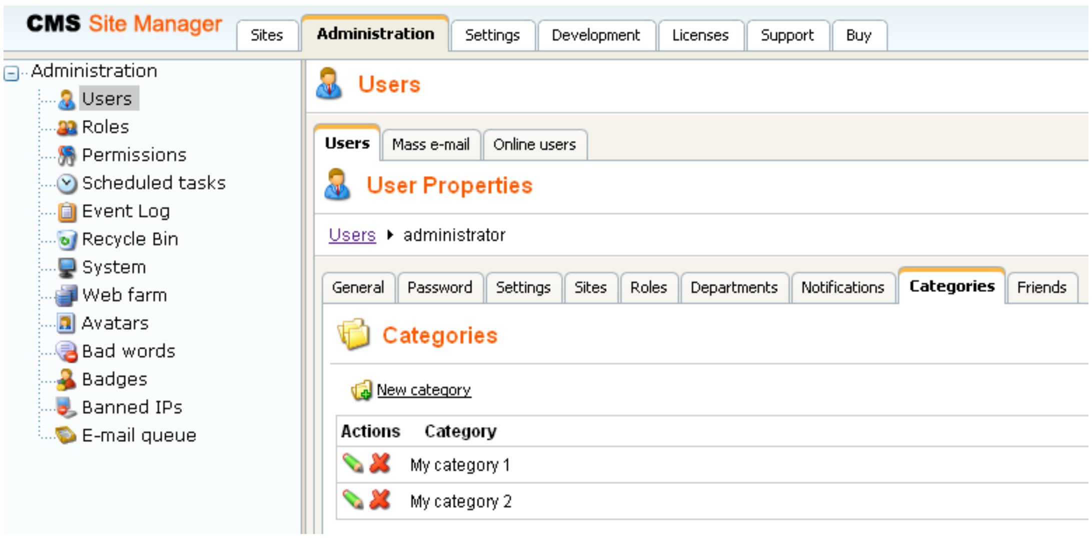
3) Click **Save**. The category has just been created and should be visible in the categories list.

## 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 its **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.

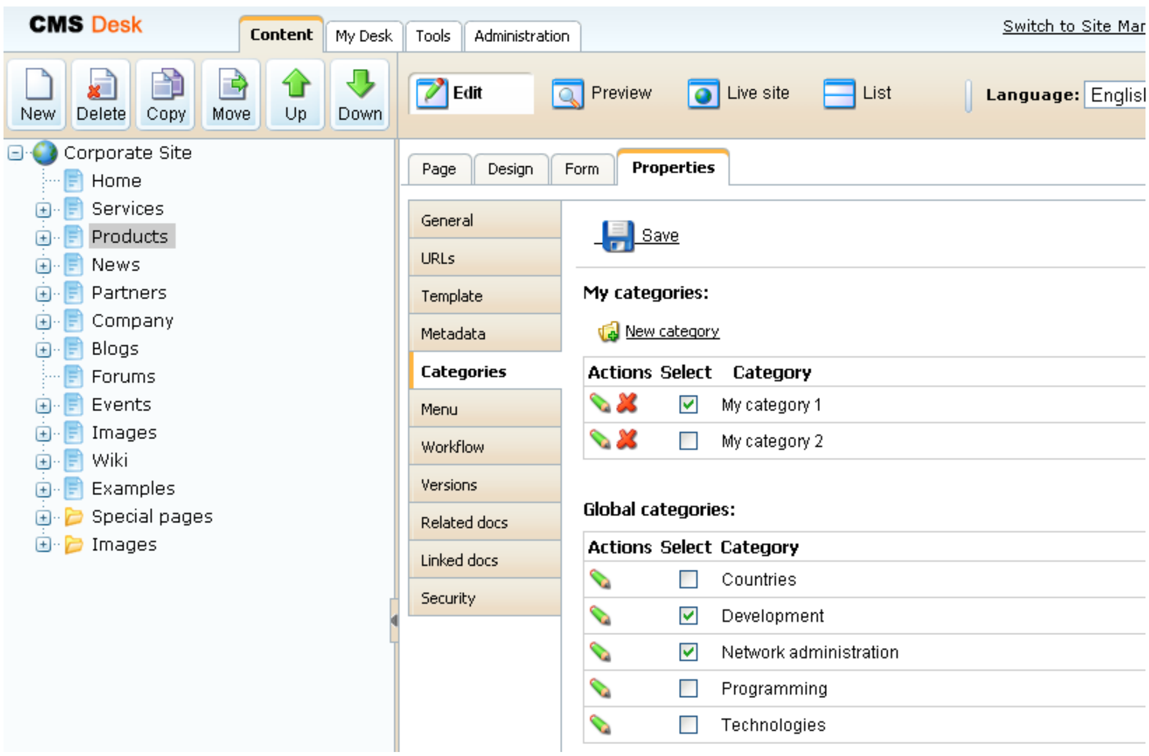


## 32.3 Adding a document to a category

Adding documents to categories is very simple.

1) Log in to **CMS Desk** and in **Edit** mode, select the document from the content tree.

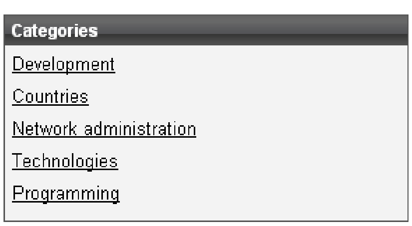
2) Switch to its **Properties -> Categories** tab.




3) In the categories lists, check the appropriate categories' **Select** check-boxes and click **Save**.

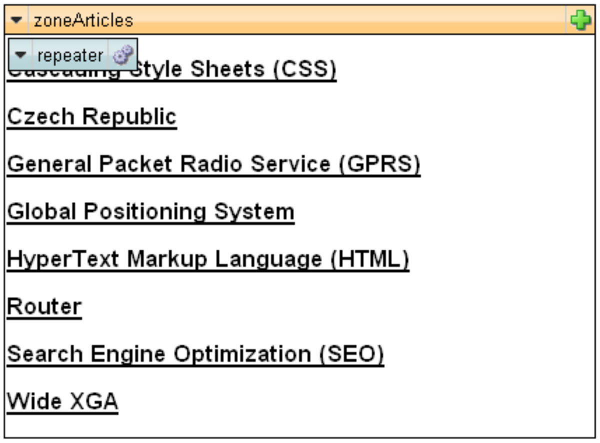
## 32.4 Using the Category list web part

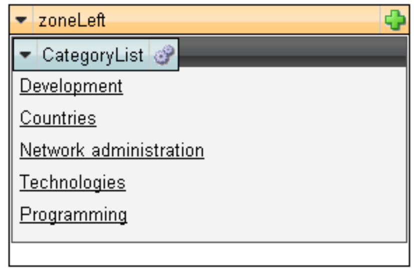
The category list web part is used to display a list of enabled global and custom categories. For each of the categories, there is a number of documents in the category displayed in brackets after the category's name. If the site visitor clicks any of the categories, she will be redirected to a page 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 as the category list, or on some other page to that the site visitor will be redirected. You can find an example of this behavior on the sample Corporate site, 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 that the repeater is placed on the same page as the category list, 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 **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?}' ))
```

# Part XXI

# 33 Module Content rating

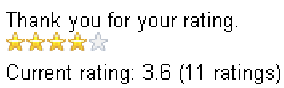
## 33.1 Overview

The content rating module can be used to give users the possibility of rating any document on your web site. The module's main part is the Content rating web part. You can place this web part on any page of your web site 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.

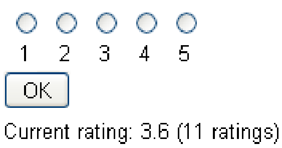
## 33.2 How to enable content rating on the web site

The content rating web part gives users the possibility of rating a document. It has three default appearance modes:

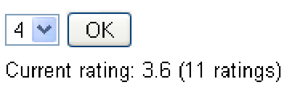
### Stars




### 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 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 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 after the user submits her rating; macros that can be used: {0} your rating, {1} overall rating, {2} overall number of votes

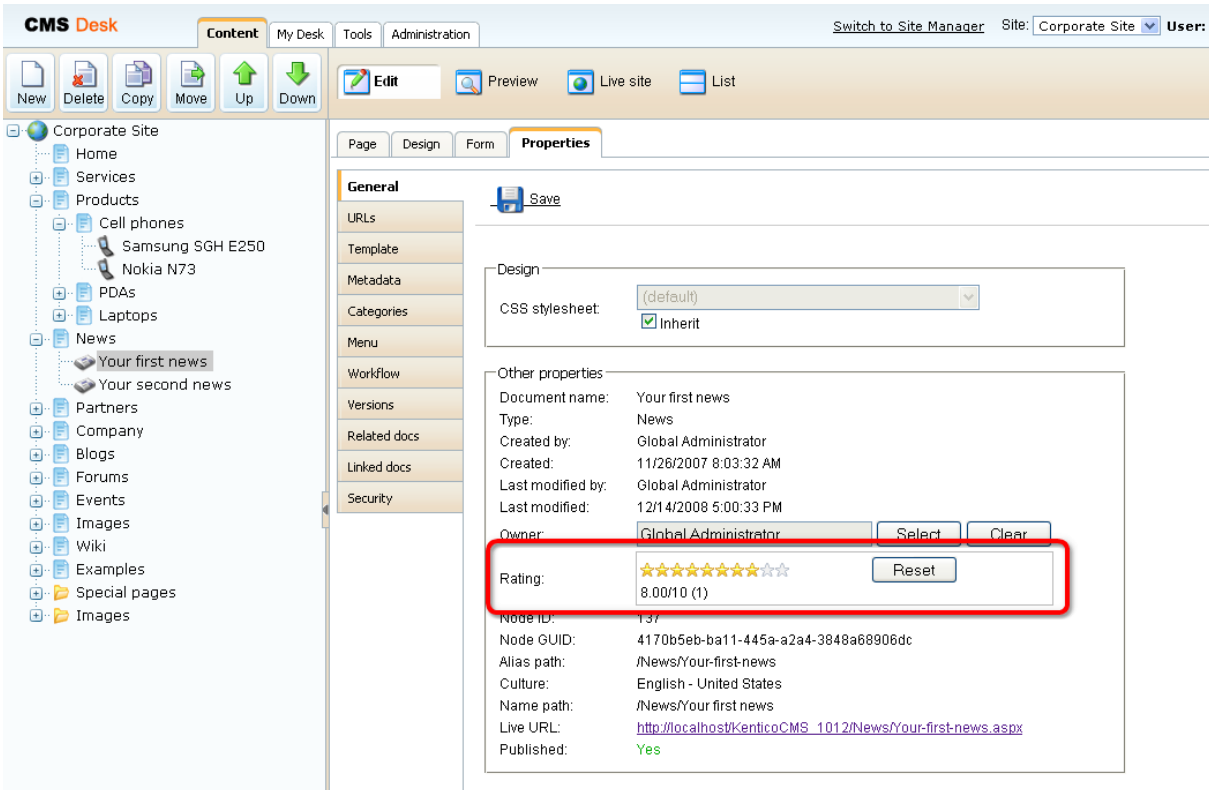
- **Check permissions** - if checked, permissions set by the Allow for public and Hide to unauthorized roles properties will be checked
- **Allow for public** - if checked, rating will be allowed to public unauthorized users; if unchecked, all users will be allowed to rate
- **Check if user rated** - 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 ratings and even cheat-rate by simply clicking the stars on this page.

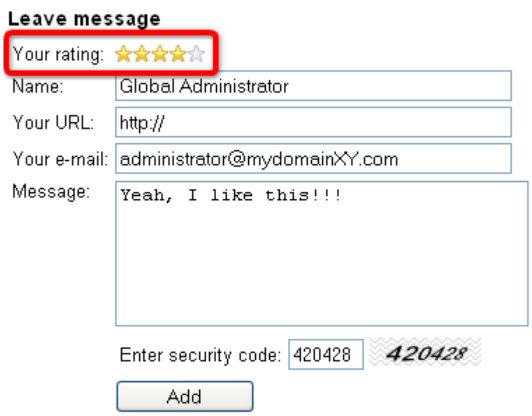


## 33.3 Other involved web parts

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.




## 33.4 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:

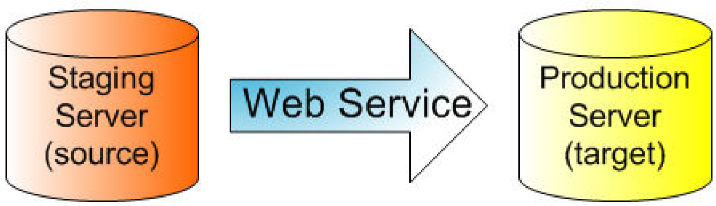
```
<%@ Register Src="~/CMSAdminControls/ContentRating/RatingControl.ascx"
TagName="RatingControl" TagPrefix="cms" %>

<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)))) %>' />
```

# Part XXI

# 34 Module Content staging

## 34.1 Overview

The **Content staging** module allows you to separate the content in development, staging (editing) and production (live) environment. It allows you to easily transfer document and object changes to another server or make a complete synchronization of documents and objects from one server to another. You can synchronize documents and objects to multiple production servers with separate databases. The synchronized data is transferred over a secured web service.




Staging module supports synchronization of the following data:

- **Document data**
- **Document attachments** – if document contains the 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
- **Global objects**

You need to ensure that both staging and production site use the same settings (document types, templates, web parts …), code files and that both servers use the same version of the CMS.

## 34.2 Content staging configuration

Configuration of the Content staging module consists of two parts: the source and target server configuration.

### Microsoft WSE Configuration

Open the **bin** folder under your web project and rename the file **Microsoft.Web.Services3.dll.rename** to **Microsoft.Web.Services3.dll**.

## Source (Staging) Server Configuration

Source configuration is required for the staging (testing) server. You need to enable content staging using the following parameters in **Site Manager -> Settings -> Content staging** section.

| | |
|---|---|
| Log content changes | If checked, the synchronization tasks are automatically generated when the content is modified. |
| Log object changes | If checked, the synchronization tasks are automatically generated when objects are modified. |

Now you need to configure the target server(s). Go to the **CMSDesk –> Tools –> Content staging -> Servers**. Here you can manage the list of target servers. Every site can use different target servers for content staging. You can even synchronize the content between the source and target server between different sites in the same web site instance (in the same database).

To add a new server, click **New server**, enter the target server properties and save them:

| | |
|---|---|
| Server display name | Server display name displayed to the users. |
| Server code name | Server name used in the code. |
| Server URL | Staging service URL that points to the content staging web service of the target server.<br><br>The web service page is located at ~/CMSPages/syncserver.asmx, so the URL should look like this:<br>http://www.targetserver.com/CMSPages/syncserver.asmx |
| Enabled | If checked, the target server is enabled, it means that the tasks are automatically generated for the server and the synchronization is enabled for this server. You can temporarily disable the server by unchecking the box in case of server maintenance or any other case. |
| Server authentication | Server authentication settings. You should set the same parameters that are configured for your target server.<br><br>The default user name is **admin** and the default password is **pass**. |

**Please note**

Tasks are logged only for created and enabled servers. If there is no server created or if no server is running, no tasks are logged.

### Target (Production) Server Configuration

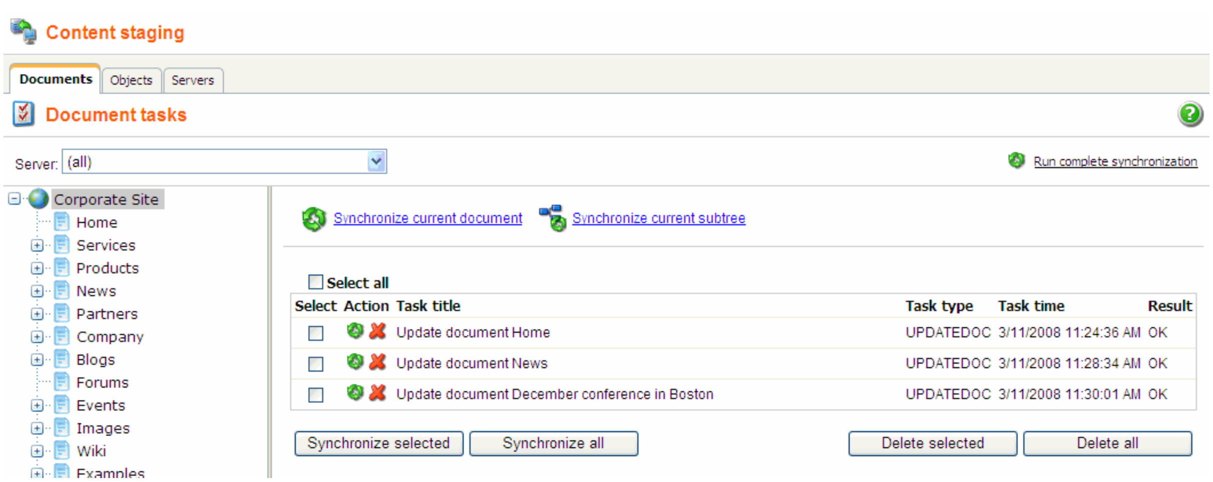
You can configure the target server synchronization web service in **Site Manager -> Settings -> Content staging**. By default, the staging web service is disabled.

| | |
|---|---|
| Enable staging service | If checked, the staging service is enabled for the given site. |
| Staging service authentication | Staging service authentication type. You can choose from the following options:<br><br>**USERNAME** – username/password authentication (fast, recommended for the data without high security requirements)<br><br>**X509** – X509 Certificate authentication (more secure, slower, requires certificates)<br><br>It's recommended that you choose to use USERNAME authentication to configure the staging first, test the synchronization and then you can optionally configure the site for X509 certificates. |
| Staging service password<br>Staging service username | Username and password for the USERNAME authentication. |
| Server key ID and Client key ID | Certificates keys for the X509 authentication, the certificates need to be stored within Local computer – Personal certificate store. |

## 34.3 Synchronizing the content

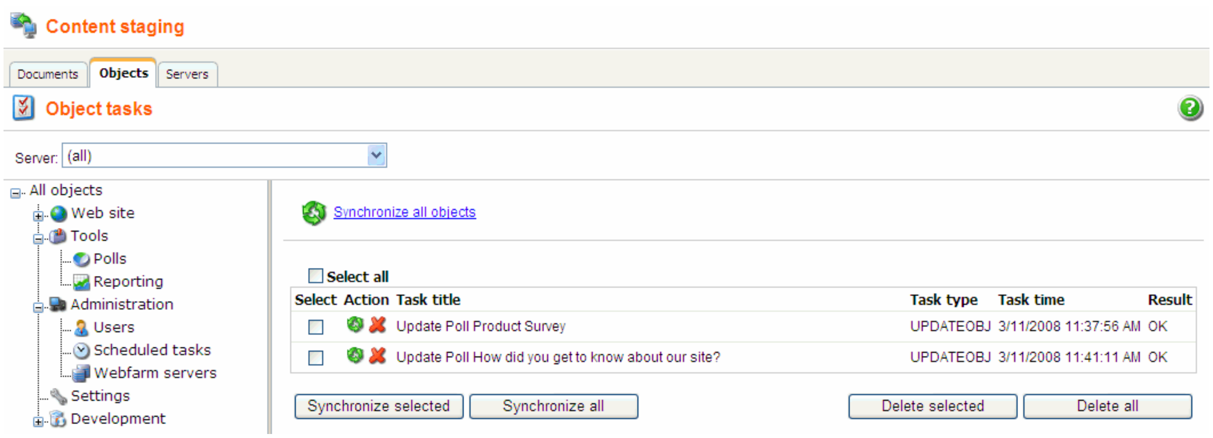
All changes made to the documents and objects are tracked in the database, in the synchronization log. You can view them in the **CMS Desk –> Tools –> Content Staging** dialog, under the **Documents** and **Objects** tabs.

In the screenshot below, you can see the **Documents** tab. By clicking the website tree root in the tree view on the left side of the screen, you can view a list of all changes  (synchronization tasks) made to all the documents of the website. By clicking a particular document, you can view only the changes made to it.



In the following screenshot, you can see the **Objects** tab. Compared to the **Documents** tab, you can't view all the changes made to all objects at once. You can view these separately, sorted by object categories or particular object types, just as listed in the tree view on the left side of the screen.

The main categories are Web site, Tools, Administration and Development. The first category contains object changes connected to the current website, whilst the other three contain object changes for global objects.




Executing both document and object synchronization tasks is done the same way, but some functionalities are available only for one of the two categories.

### Choosing target servers to be synchronized

You can choose to synchronize only selected or all target servers using the **Server** drop-down list.

### Running complete synchronization (documents only)

If you click the **Run complete synchronization** link, all documents are synchronized to the target server(s). In case you made any changes to content on the target servers in the meantime, these changes will be overwritten. Extra documents will not be deleted on the target server. This option synchronizes all documents even if there are no synchronization tasks listed.

### Synchronizing current document or current sub-tree (documents only)

These options synchronize the selected document/all documents in the sub-tree, even if there are no synchronization tasks listed.

### Synchronizing all objects (objects only)

Performs all synchronization tasks for all objects and then synchronizes all objects, even if there are no synchronization tasks listed for them.

### Synchronizing selected tasks

You can choose the document/object synchronization tasks to be executed using the check-boxes in the list and then clicking the **Synchronize selected** button. By clicking the **Synchronize all** button, all listed synchronization tasks will be performed, no matter which of them are currently selected.

### Deleting synchronization tasks

If you do not wish to synchronize some changes, you can delete them using the **Delete selected** or

**Delete all** buttons. The first one deletes the tasks selected by the check-boxes in the list, the other one deletes all listed tasks, no matter which of them are currently selected.

**When the task fails**

You can see the reason why some of the synchronization tasks failed by clicking the **Failed** link in the **Result** column.

## 34.4 Using X.509 authentication

In order to use **X.509** authentication, you need to install your own, or our 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 (for the client certificate "Client private.pfx") and on the target server (for the server certificate "Server private.pfx"):

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 .pfx file in folder **C:\Program Files\Kentico CMS\<version>\SampleCertificates**.

8. Enter the following password for the sample certificates (it is the same for client and server certificate): **wse2qs**

9. Now you need to grant the ASPNET (for Windows 2000 and Windows XP) or (for NT Authority\Network Service in Windows 2003) account with **READ** permissions for the certificate file. You can do that using the **WseCertificate3.exe** tool that can be found in folder **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 user ASPNET (for Windows 2000 and Windows XP) or the user NT Authority\Network Service (for Windows Server 2003) with **READ** permission for this file.

If you're using your own certificates (highly recommended), you will need to update the following values in the **Site Manager -> Settings -> Content Staging** dialog:

- Client key ID
- Server key ID

Here's how to get the ID's:

You will use the WseCertificate3.exe tool located in folder **C:\Program files\KenticoCMS\<version>\SampleCertificates** to get the certificate keys.
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), 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.

## 34.5 Using API to synchronize documents

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#]**

```
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#]**

```
// 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);
     }
   }
 }
```

## 34.6 Configuration of automatic synchronization

Automatic synchronization is available only for synchronizing 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 interval.

## 34.7 Security

You can control the access to the Content staging module in **Administration -> Permissions ->** select the **Modules -> Content staging** permissions matrix.

**Permissions**

Site: Corporate Site
Permission type: Modules
Permission matrix: Content staging

| | Display module | Manage object tasks | Manage servers | Manage synchronization tasks |
|---|---|---|---|---|
| CMS Designers | ☐ | ☐ | ☐ | ☐ |
| CMS Desk Administrators | ☑ | ☐ | ☑ | ☑ |
| CMS Editors | ☐ | ☐ | ☐ | ☐ |
| CMS Public Users | ☐ | ☐ | ☐ | ☐ |
| CMS Readers | ☐ | ☐ | ☐ | ☐ |
| Everyone | ☐ | ☐ | ☐ | ☐ |
| Gold Partners | ☐ | ☐ | ☐ | ☐ |
| Registered Users | ☐ | ☐ | ☐ | ☐ |
| Silver Partners | ☐ | ☐ | ☐ | ☐ |

The module recognizes the following permissions:

- **Display module -** enables the module to be visible in the user interface
- **Manage servers** - allows user to manage target server configurations
- **Manage object tasks** - allows user to synchronize documents content or delete document synchronization tasks
- **Manage synchronization tasks -** allows user to synchronize objects or delete object synchronization tasks

# Part XX

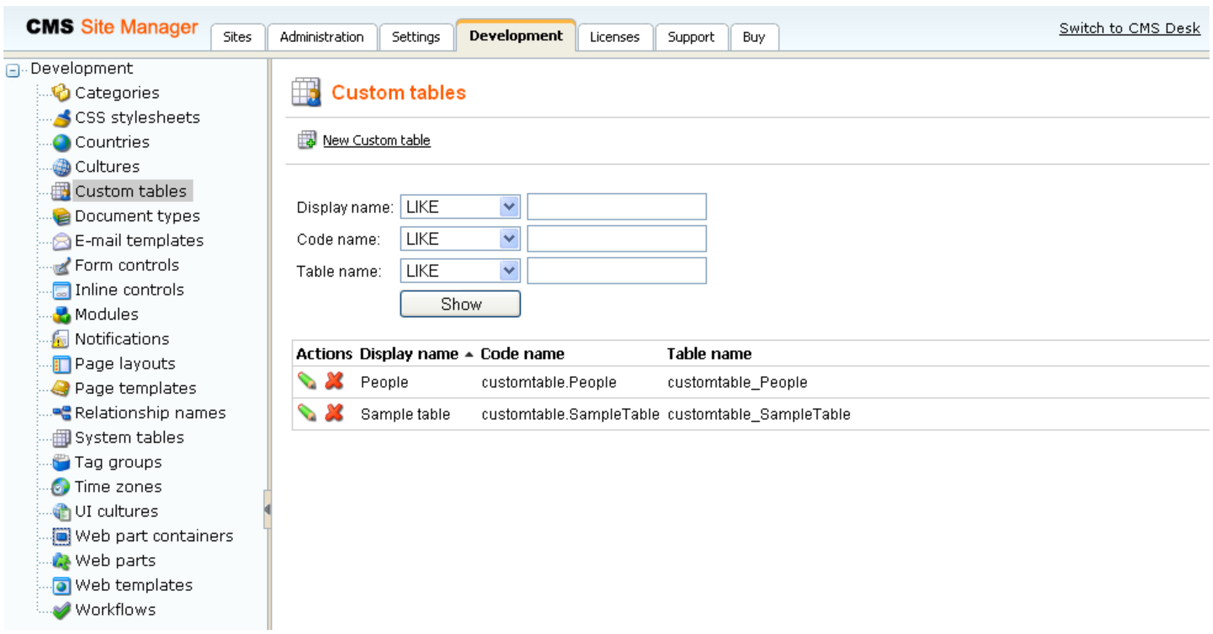
# 35 Module Custom tables

## 35.1 Custom tables module overview

The Custom tables module allows users to store data in their own custom database tables without the need to use the tree structure, which is inefficient for larger amounts of data.

## 35.2 Managing custom tables

The administration interface for custom tables management is located in **Site Manager -> Development -> Custom tables**.




The top part of the page is a filter. You can filter displayed custom tables by Display name, Code name and Table name. You can also **Edit** () or **Delete** () the listed records. Creating a new custom table is also done here and is described in the following example:

### New Custom table wizard

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. Their first names, last names and dates of birth will be stored in the table.

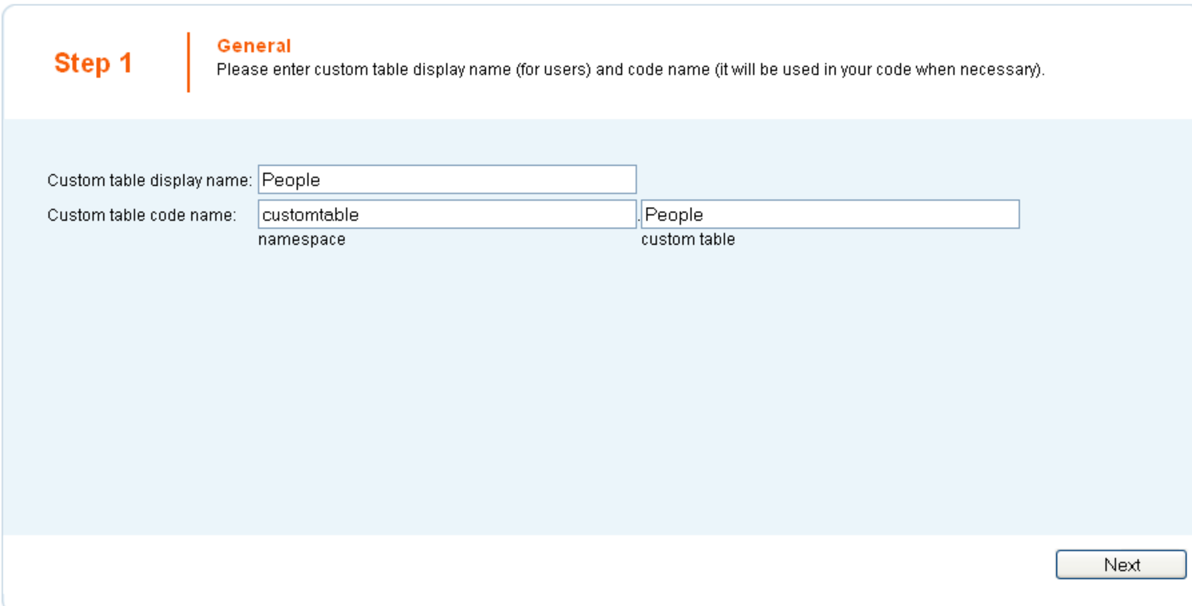
1) Go to **Site Manager -> Development -> Custom tables**.

2) Click the **New Custom table** link.

3) The New Custom table wizard starts. On the first step of the wizard, you are asked to fill in the custom table's display name and code name. 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

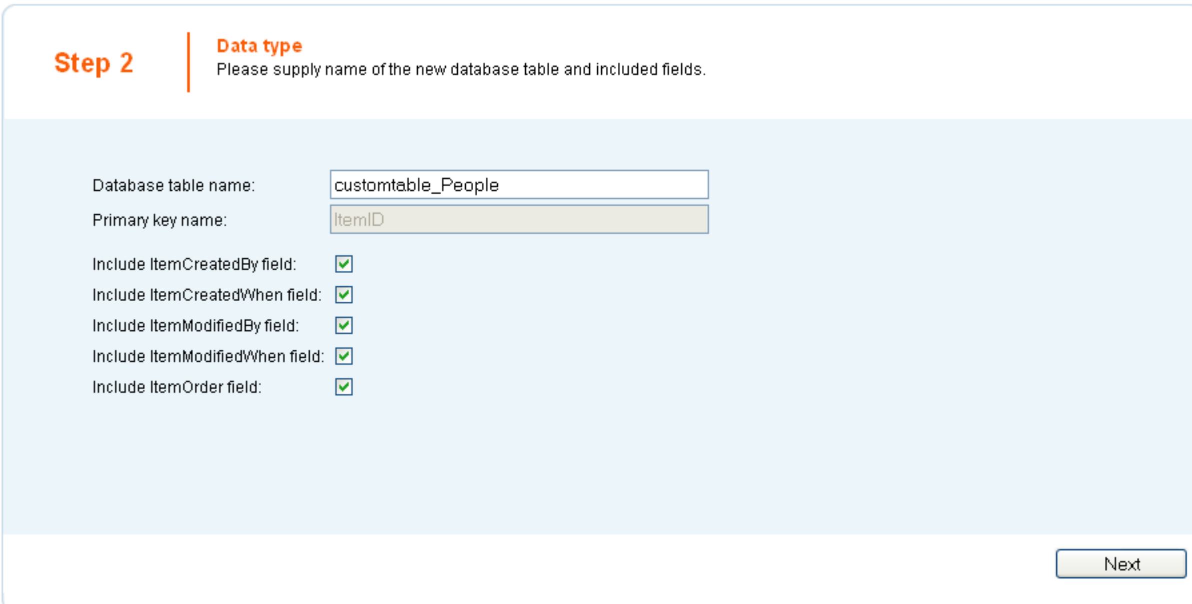
and click **Next** to continue.



4) In the second step, you are asked to fill in the database table name. A name in the format <namespace>_<code name> is pre-filled automatically. The primary key name cannot be changed by default. Using the check-boxes below, you can determine which of the default fields should be included in the table. The meanings of the fields are the following:

- **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

Leave default values for all parameters and click **Next** to continue.



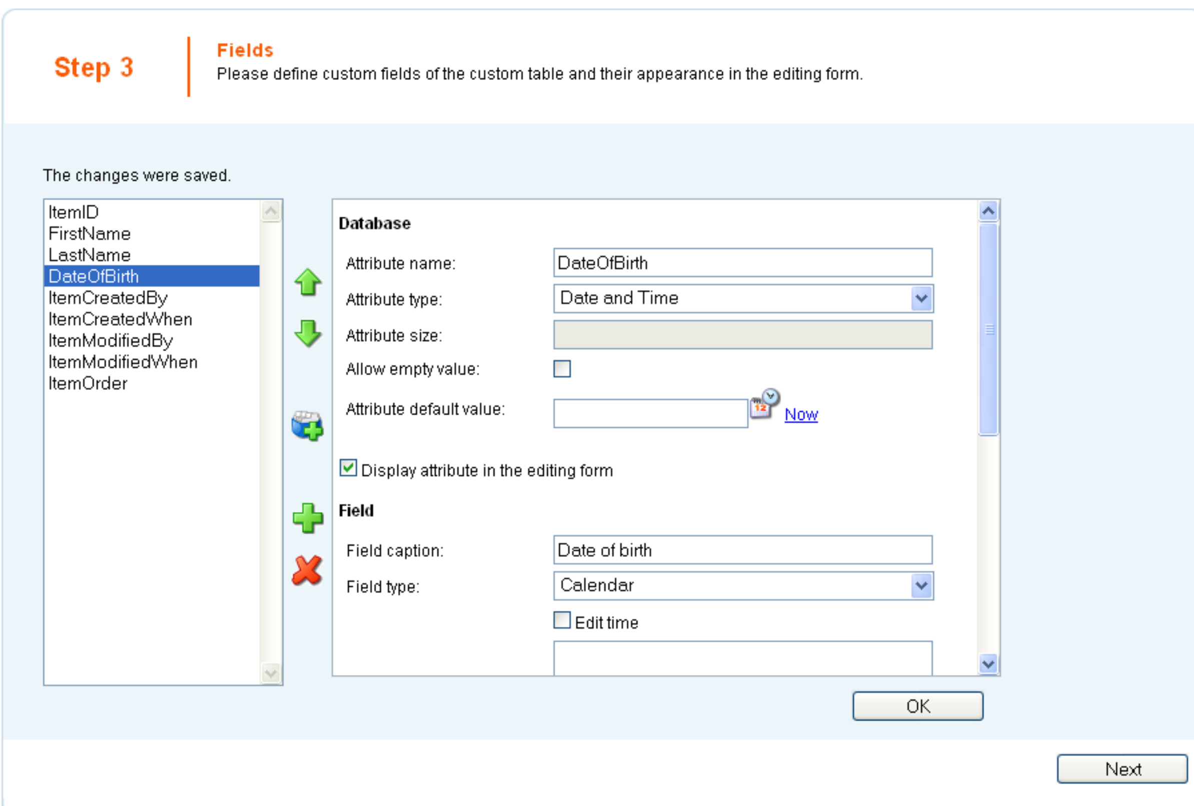
5) In the third step, the field editor that you should be already familiar with is displayed. Create three fields with the following parameters:

- **Attribute name:** FirstName
- **Attribute type:** Text
- **Attribute size:** 100
- **Field caption:** First name
- **Field type:** Text box

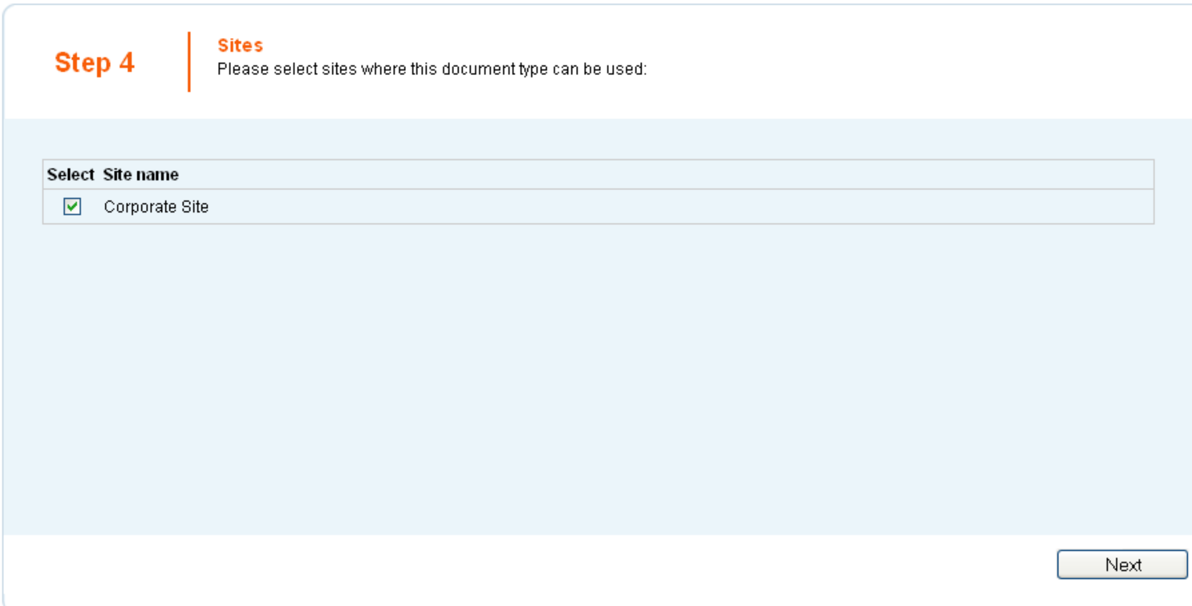
- **Attribute name:** LastName
- **Attribute type:** Text
- **Attribute size:** 100
- **Field caption:** Last name
- **Field type:** Text box

- **Attribute name:** DateOfBirth
- **Attribute type:** Date and Time
- **Field caption:** Date of birth
- **Field type:** Calendar
- **Edit time:** unchecked

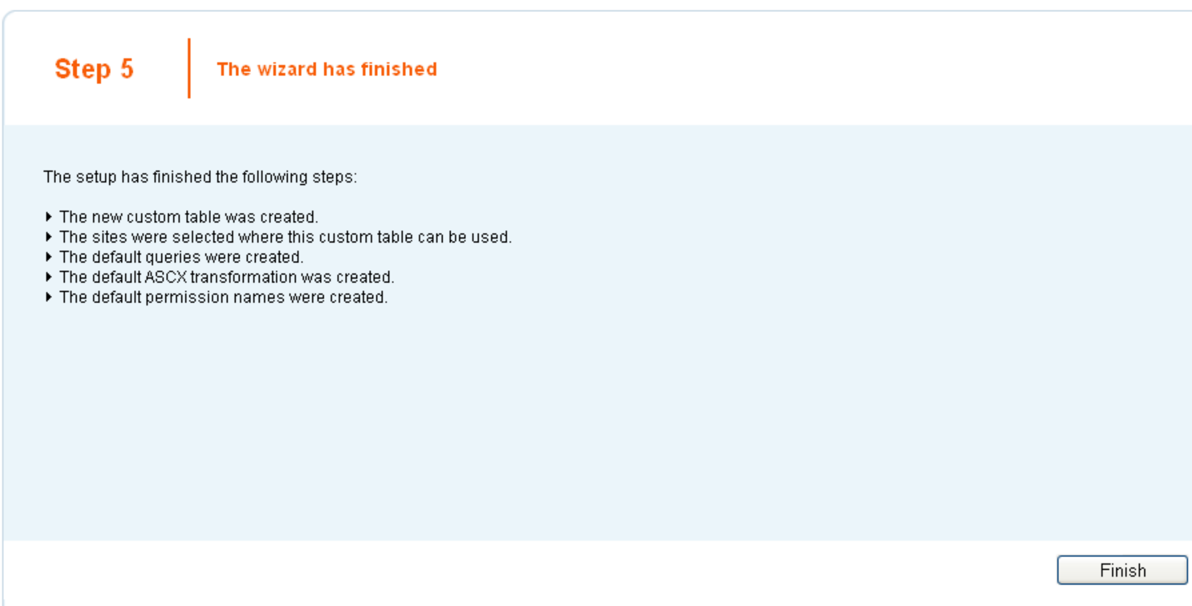
When you are finished, click **Next** to continue to the following step.



6) On Step 4, you only have to select by which sites will the custom table be used. Check the **Corporate Site** check-box and click **Next**.

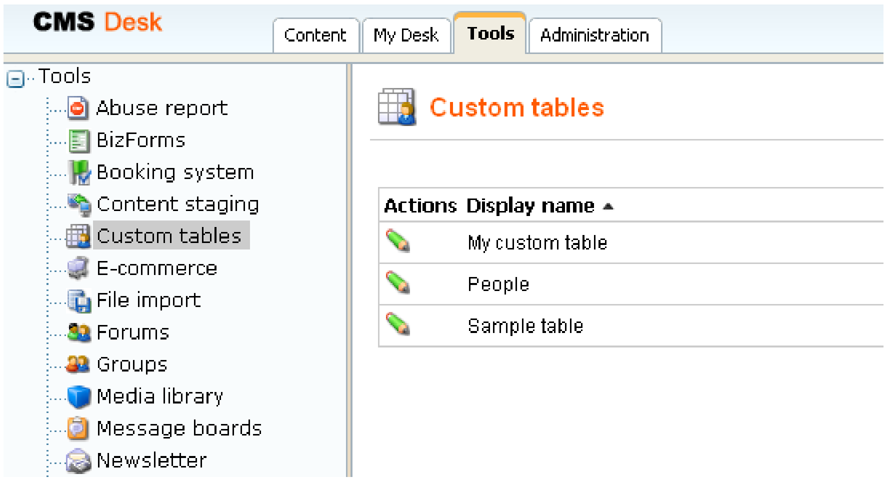



7) The fifth step gives you an overview of what has been done automatically. To finish the wizard, click **Finish**. You will be redirected back to the list of custom tables in **Site Manager -> Development -> Custom tables**.



## 35.3 Adding items into custom tables

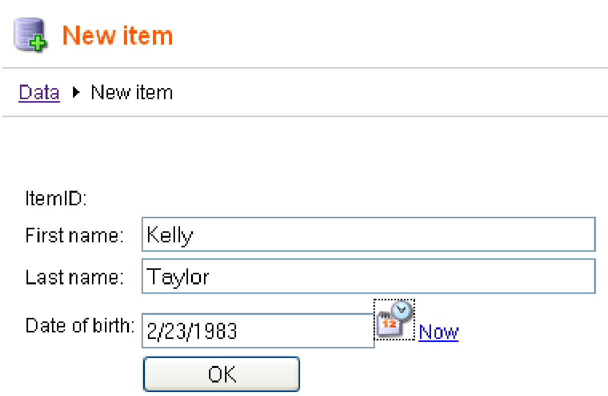
The administration interface for adding data into custom tables is located in **CMS Desk -> Tools -> Custom tables**.




In the following example, you will learn how to add data into the sample **People** custom table we have created in the previous chapter (read here):

1) Click the **Edit** ( ) icon next some of the listed tables. 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.

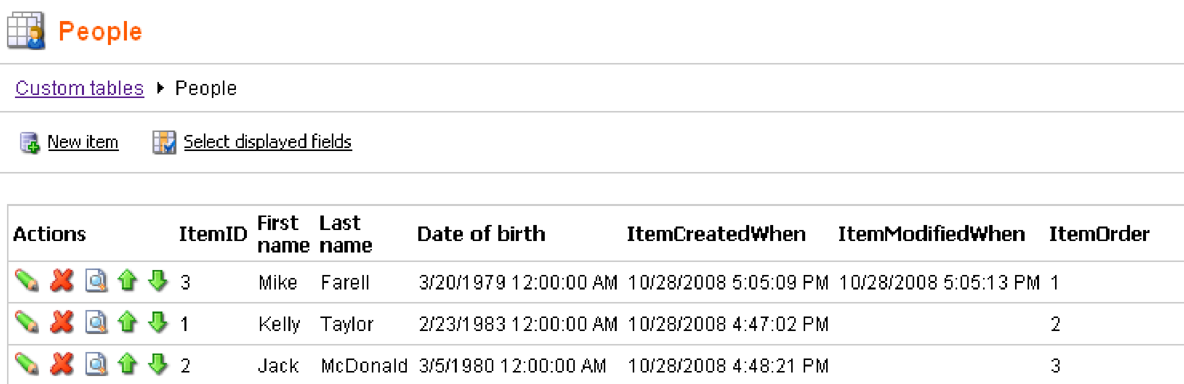

2) Click the **New item** link. Form that you can see in the following screenshot will be displayed. Enter some sample data and click **OK**.




3) The data you entered has just been saved into the custom table. You can edit it now or use the Create another link to create another record. Try creating at least two other records.

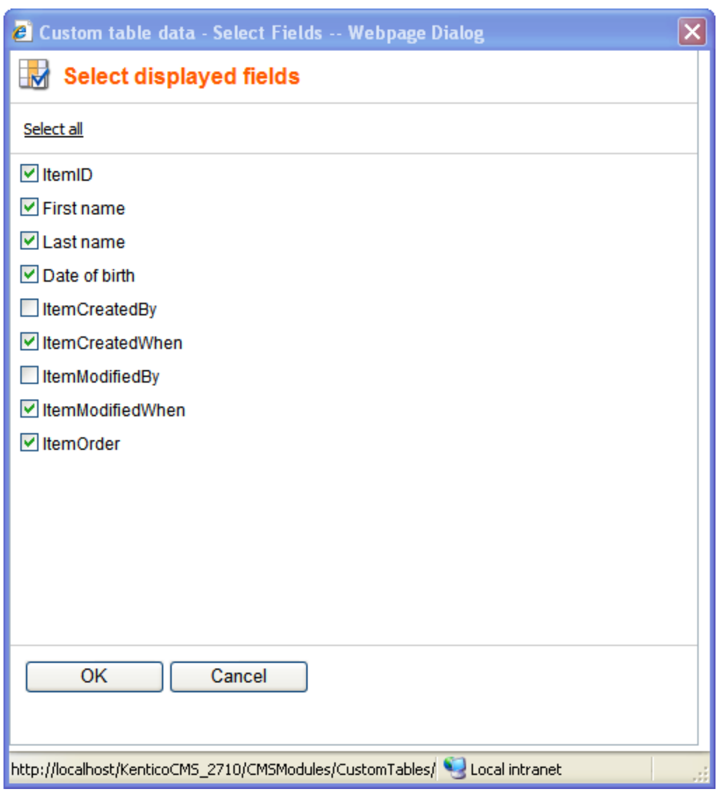
4) When you're finished, switch back to the **Custom tables** list in **CMS Desk -> Tools** and choose to

**Edit** () the **People** custom table. You should see the list of entries where the blank page was before entering the records.

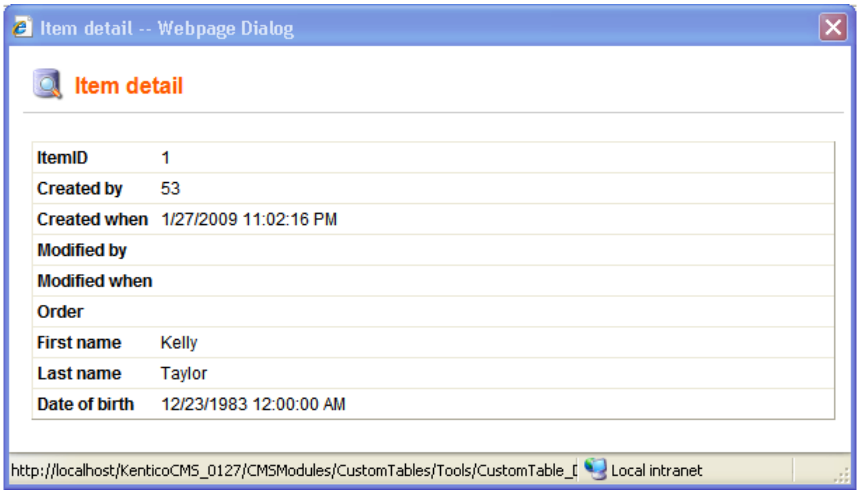



| Actions | ItemID | First name | Last name | Date of birth | ItemCreatedWhen | ItemModifiedWhen | ItemOrder |
|---|---|---|---|---|---|---|---|
| | 3 | Mike | Farell | 3/20/1979 12:00:00 AM | 10/28/2008 5:05:09 PM | 10/28/2008 5:05:13 PM | 1 |
| | 1 | Kelly | Taylor | 2/23/1983 12:00:00 AM | 10/28/2008 4:47:02 PM | | 2 |
| | 2 | Jack | McDonald | 3/5/1980 12:00:00 AM | 10/28/2008 4:48:21 PM | | 3 |

5) You can choose which fields are to be displayed 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 and which are not. When you make the selection and click **OK**, you should see the result immediately.



6) If you click the **View** () icon next to each of the 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 list:




7) You can also order the records in the list using the **Up** () and **Down** (). The order will then be stored in the **ItemOrder** property of each of the records.

8) Records in the custom table can also be **Edited** () and **Deleted** () on this page.

## 35.4 Available web parts

The Custom tables module comes with three web parts that can be used for displaying custom tables' content. You can find detailed descriptions of each web part's properties in **Kentico CMS Web Parts** reference.

### Custom table datagrid

| FirstName ▲ | LastName | DateOfBirth |
|---|---|---|
| Jack | McDonald | 3/5/1980 12:00:00 AM |
| Kelly | Taylor | 2/23/1983 12:00:00 AM |
| Mike | Farell | 3/20/1979 12:00:00 AM |

### Custom table datalist

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 for custom tables in chapter Transformations for custom tables.

| | | | |
|---|---|---|---|
| First name: | Kelly | First name: | Jack |
| Last name: | Taylor | Last name: | McDonald |
| Date of birth: | 2/23/1983 12:00:00 AM | Date of birth: | 3/5/1980 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 for custom tables in chapter Transformations for custom tables.

| | |
|---|---|
| First name: | Kelly |
| Last name: | Taylor |
| Date of birth: | 2/23/1983 12:00:00 AM |
| First name: | Jack |
| Last name: | McDonald |
| Date of birth: | 3/5/1980 12:00:00 AM |
| First name: | Mike |
| Last name: | Farell |
| Date of birth: | 3/20/1979 12:00:00 AM |

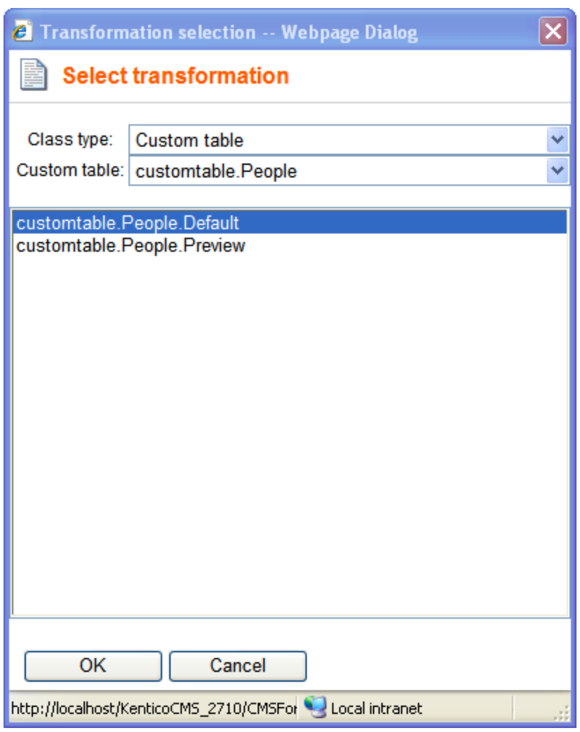
## 35.5 Transformations for custom tables

Use of transformations is essential for the custom table web parts. In the following example, you will learn how to modify the way data is displayed by changing the transformation code. You can find examples of usage of the web parts on the sample Corporate Site under **Examples -> Webparts -> Custom tables**. We will use the **Custom table repeater** example that can be found on the likely named page. We will also use the **People** custom table that we have created (read here) and populated with some records (read here) in the previous chapters.

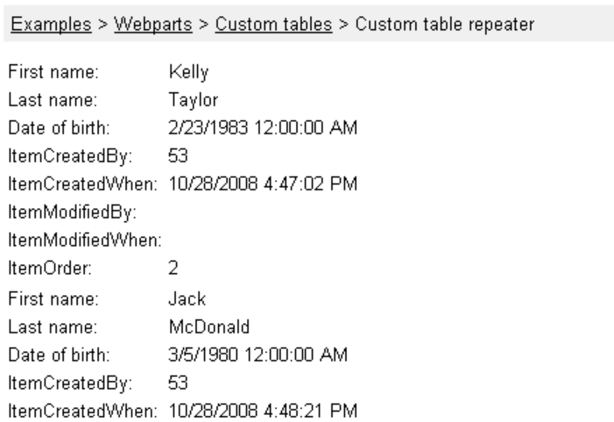
1) Go to **CMS Desk**, switch to **Edit** mode and select **Examples -> Webparts -> Custom tables -> Custom table repeater**. The repeater is bound to the **Sample table** that is a part of the Corporate site by default. It should appear as in the following screenshot:


2) Switch to the **Design** mode and choose to **Configure** ( ) the repeater's properites. First, select the custom table from that the data will be displayed. It is done by the **Custom table** drop-down list. Choose the **People** custom table.

3) Now it's time to set the used 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 - the **Default** transformation and the **Preview** transformation. Choose the Default transformation and click **OK**. Click **OK** again in the **Web part properties** window.



4) Now if you switch to the Live site, 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 unformated on your real web site, you would want 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 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 formated way, as you can see below.

Examples > Webparts > Custom tables > Custom table repeater

| | |
|---|---|
| **First name:** | Kelly |
| **Last name:** | Taylor |
| **Date of birth:** | 2/23/1983 12:00:00 AM |

| | |
|---|---|
| **First name:** | Jack |
| **Last name:** | McDonald |
| **Date of birth:** | 3/5/1980 12:00:00 AM |

## 35.6 Security

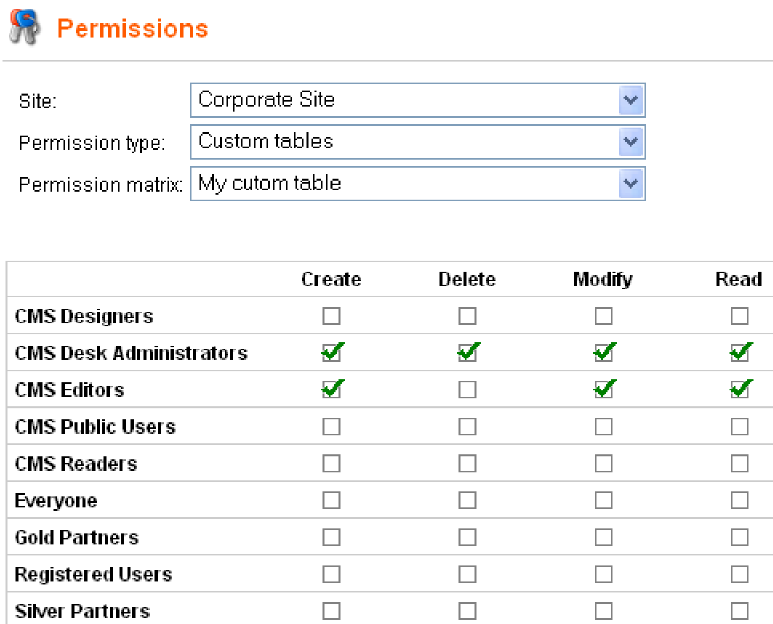
Permissions of the Custom tables module can be set in **Site Manager** or **CMS Desk** in the **Administration -> Permissions** section, in the **Modules -> Custom tables** permission matrix:


| | Create | Delete | Display module | Modify | Read |
|---|---|---|---|---|---|
| CMS Designers | ☐ | ☐ | ✓ | ☐ | ✓ |
| CMS Desk Administrators | ✓ | ✓ | ✓ | ✓ | ✓ |
| CMS Editors | ☐ | ☐ | ✓ | ☐ | ✓ |
| CMS Public Users | ☐ | ☐ | ☐ | ☐ | ☐ |
| CMS Readers | ☐ | ☐ | ☐ | ☐ | ☐ |
| Everyone | ☐ | ☐ | ☐ | ☐ | ☐ |
| Gold Partners | ☐ | ☐ | ☐ | ☐ | ☐ |
| Registered Users | ☐ | ☐ | ☐ | ☐ | ☐ |
| Silver Partners | ☐ | ☐ | ☐ | ☐ | ☐ |

The following permissions can be assigned to the particular user roles:

- **Create** - users are allowed to create data in any custom table
- **Delete** - users are allowed to delete data in any custom table
- **Display module** - makes the module visible in the user interface
- **Modify** - users are allowed to modify data in any custom table
- **Read** - users are allowed to read any custom table data

You can also set permissions for particular custom tables. This is also done in the **Administration -> Permissions** section of both **Site Manager** and **CMS Desk**. 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 following permissions can be set for each custom table:

- **Create -** users are allowed to create new records into the table
- **Delete** - users are allowed to delete records from the table
- **Modify -** users are allowed to modify existing records in the table
- **Read** - users are allowed to read data stored in the custom table

## 35.7 API examples

The following code examples show how you can handle custom tables data using Kentico CMS API.

**Deleting an item from a custom table**

**[C#]**

```
string customTableClassName = "custom.SampleTable";

// Get data class using custom table name
DataClassInfo customTableClassInfo = DataClassInfoProvider
.GetDataClass(customTableClassName);

// Check if data class info exists
if (customTableClassInfo != null)
{
        // 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 delete
        int itemId = 1;

        // Get custom table item with given item ID
        CustomTableItem item = ctiProvider.GetItem(itemId,
customTableClassInfo.ClassName);

        // Check if item exists
        if (item != null)
        {
                // Delete item
                item.Delete();
        }
}
```

**Adding an item into a custom table**

**[C#]**

```
string customTableClassName = "custom.SampleTable";

// Get data class using custom table name
DataClassInfo customTableClassInfo = DataClassInfoProvider
.GetDataClass(customTableClassName);
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, ctiProvider);

// Set value of a custom table item field
item.SetValue("TestField", "Sample item");

// Insert the item
item.Insert();
```

### Editing an item in a custom table

**[C#]**

```
string customTableClassName = "custom.SampleTable";

// Get data class using custom table name
DataClassInfo customTableClassInfo = DataClassInfoProvider
.GetDataClass(customTableClassName);
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.ClassName);

// Set value of the custom table item field
item.SetValue("TestField", "Sample item");

// Update item
item.Update();
```

### Getting items from a given custom table

**[C#]**

```
string customTableClassName = "custom.SampleTable";

// Get data class using custom table name
DataClassInfo customTableClassInfo = DataClassInfoProvider
.GetDataClass(customTableClassName);
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());

// Get custom table items
DataSet dsItems = ctiProvider.GetItems(customTableClassInfo.ClassName, null, null);

// Check if DataSet is not empty
if (!DataHelper.DataSourceIsEmpty(dsItems))
{
        // Handle the retrieved data
}
```

# Part XXV

# 36 Module E-commerce

## 36.1 Overview

For more information about the **E-commerce** module please refer to **Kentico CMS E-commerce Guide**.

# Part XXV

# 37 Module E-mail queue

## 37.1 Overview

The E-mail queue module was designed to enable sending of large amounts of e-mails, e.g. when sending newsletter issues to the subscribers, etc. The e-mails in the queue are sent **automatically**, no manual sending is necessary as the E-mail queue sender scheduled task handles it. When an e-mail is not sent successfully, it remains in the queue so that it can be resent later.

## 37.2 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:

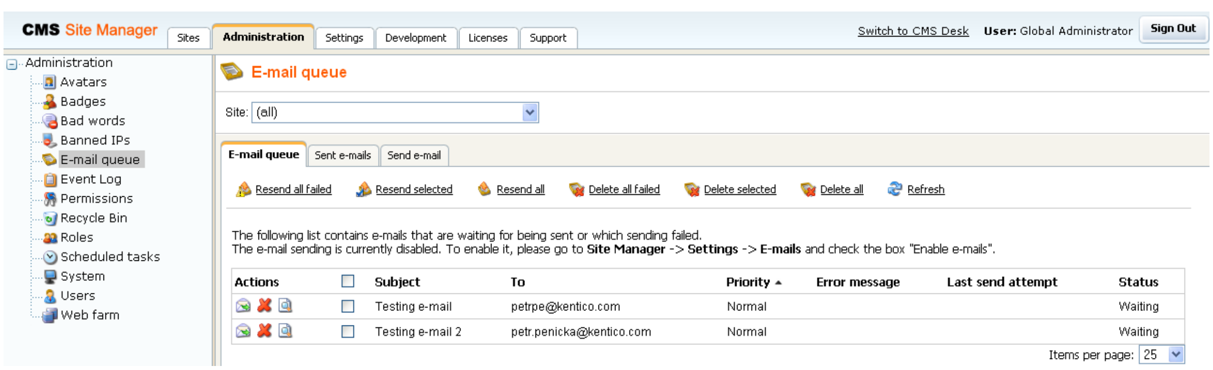
### E-mail queue tab

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 not successfully 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

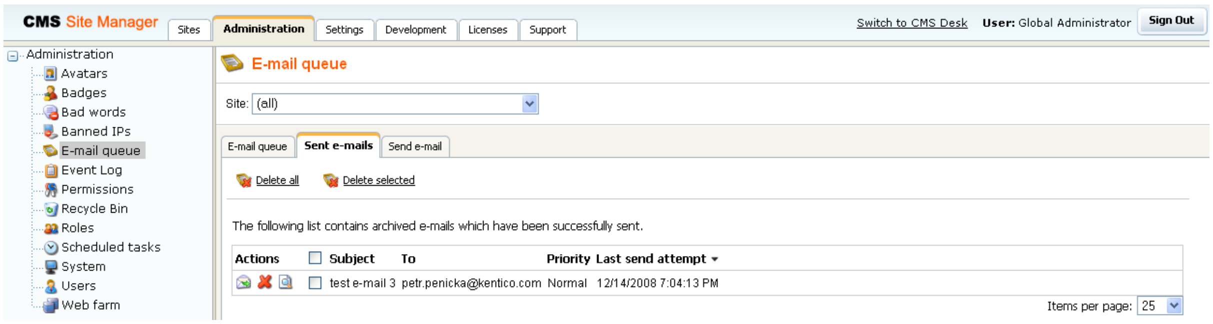



### Sent e-mails tab

This tab displays a list of e-mails that have been successfully sent via the e-mail queue. You can set how long will the e-mails stay in this list by 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 two 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




### Send e-mail tab

From this tab, you can easily send a single e-mail to a specified recipient (not only the site users).

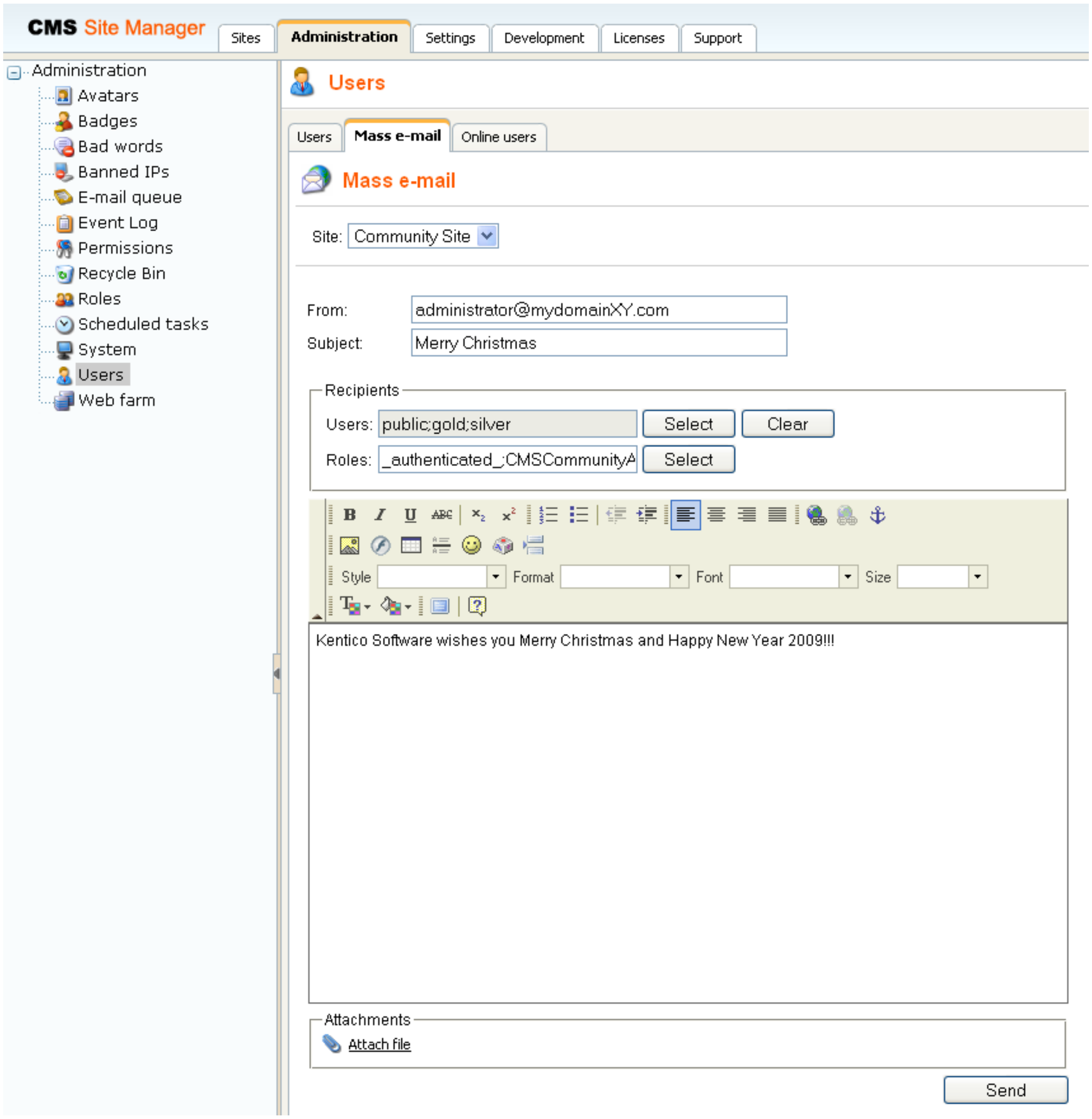
## 37.3 Sending mass e-mails

As e-mail queue allows 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 user.

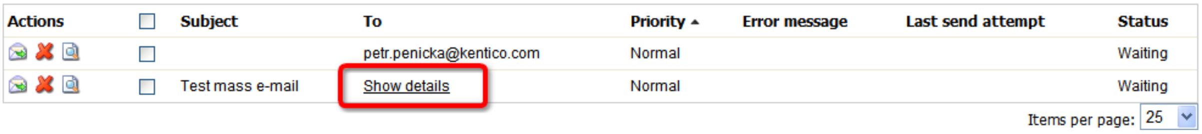
You can do this in **Site Manager -> Administration -> Users -> Mass e-mail**.

Using the **Site** drop-down list, you can select which site are the recipients related to. Based on this choice, users and user roles related to the chosen site can be selected using the **Select** buttons in the **Recipients** section. When choosing a user or role, *CTRL+left click* can be used to select multiple users or roles. If you choose **(all)** from the Site drop-down list, the Roles field will be hidden as user roles are always site-related.

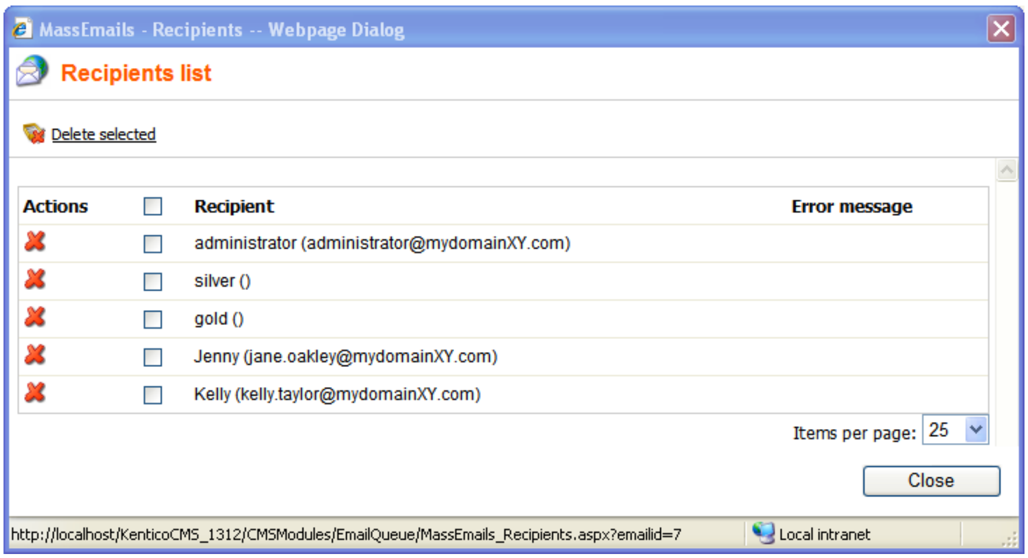
You can also add some attachment to the mail using the **Attach file** link.



In the e-mail queue, you will see the **Show details** link instead of recipient's e-mail address for mass e-mails, as you can see in the screenshot below.



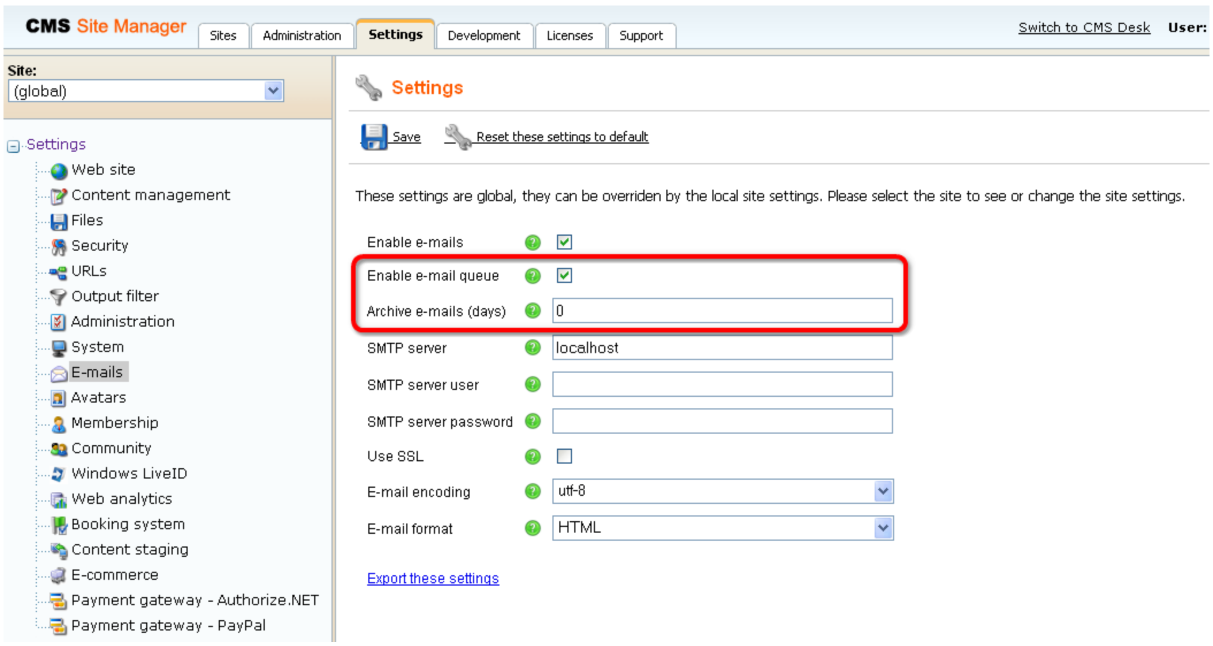

After clicking the link, a window will appear as in the screenshot below. You can delete recipients from the list, so that the mass e-mail will not be delivered to them. This can be done using the **Delete** ( ) icons or by selecting more recipients using the check-boxes and clicking the **Delete selected** link.



## 37.4 Settings

E-mail settings are located in **Site Manager -> Settings -> E-mails**. Among other e-mail related settings, these settings are related to the e-mail queue:

**Enable e-mail queue** - if checked, e-mail queue will be used while sending e-mails.
**Archive e-mails (days)** - number of days for that will e-mails sent via e-mail queue remain stored on the Sent e-mails tab

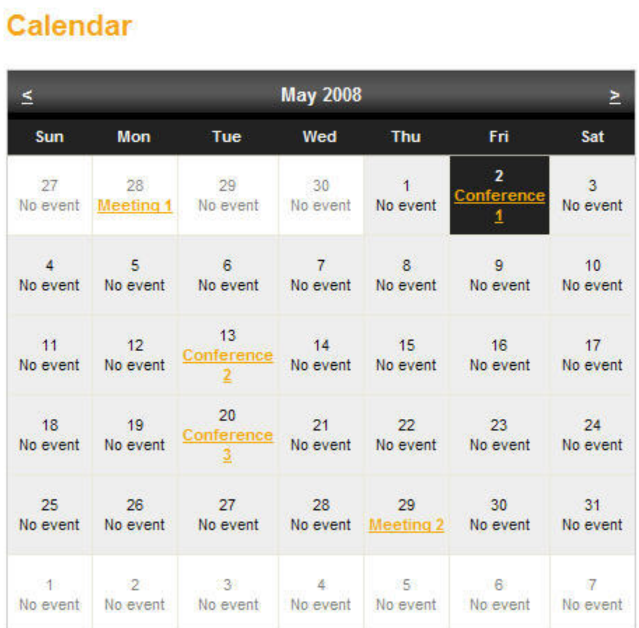


# Part XXV

# 38 Module Event calendar

## 38.1 Event calendar module 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.




## 38.2 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** mode. 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**.

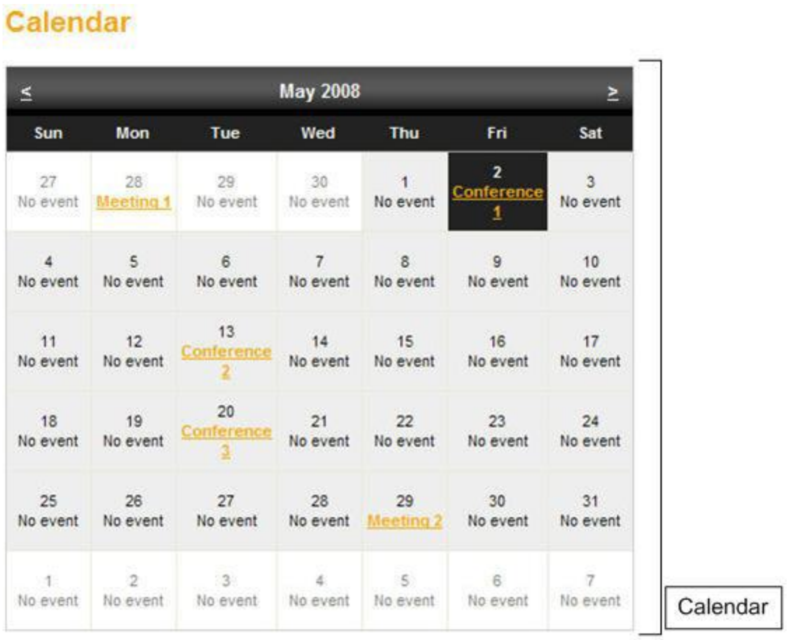
The web part properties dialog appears. Here is a list of properties specific for this web part:

| | |
|---|---|
| **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 web site from that you want to display the content. If left empty, the current web site 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.

# 38.3 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:



# Part XXI

# 39 Module File import

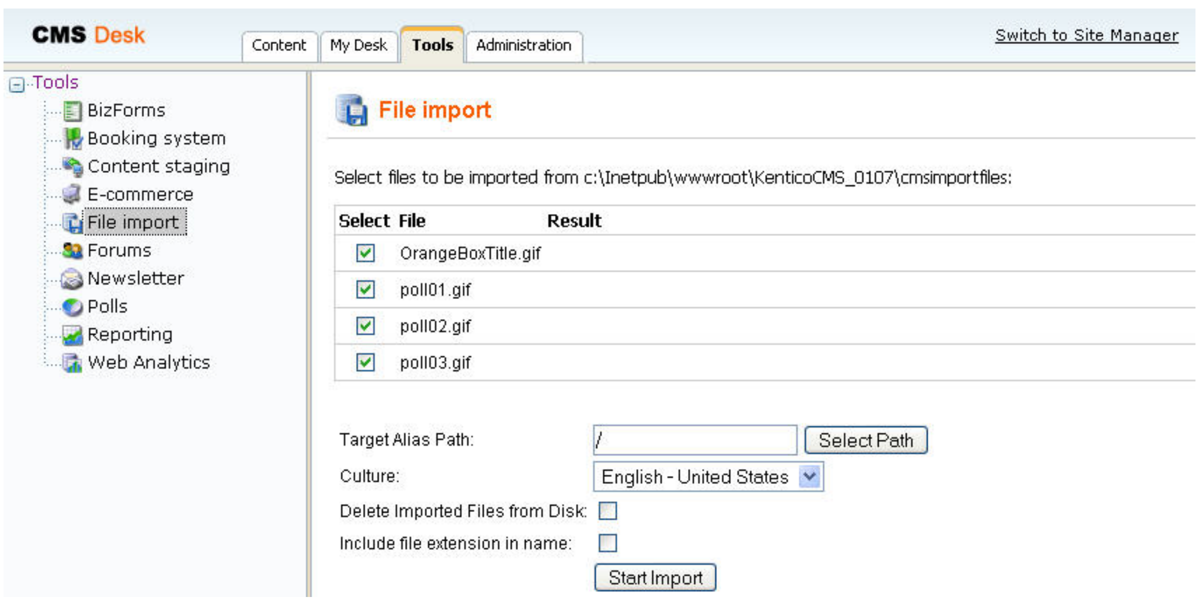
## 39.1 File import

The file import module allows you to import many files including their folder structure from the disk to Kentico CMS content repository, so that you do not have to upload them one-by-one using the user interface.

You can find the File import module in **CMS Desk -> Tools -> File import**. The label at the top displays the folder where the files are imported from. If the folder doesn't exist, you may need to create it on the disk.

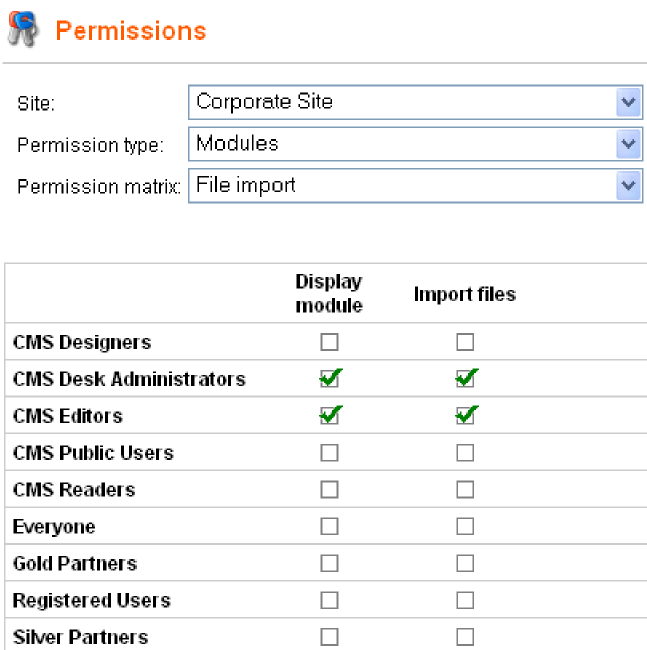
You can also configure a custom file import path in **Site Manager -> Settings -> Files -> File import folder**.

After you upload the files to the given folder, you may need to refresh the page by clicking **File import** again. Now you should see the list of files with checkboxes that allow you to choose which files should be imported. You can also choose where the files will be stored in the content tree and if you want to delete the files from the disk once they are imported.



## 39.2 Security

The access to the file import module can be controlled in **CMS Desk -> Administration -> Permissions**, after you select the **Modules -> File import** permission matrix:




The following permissions can be set:

- **Display module** - the module will be displayed in the user interface
- **Import files** - importing files will be allowed to users belonging to the user group

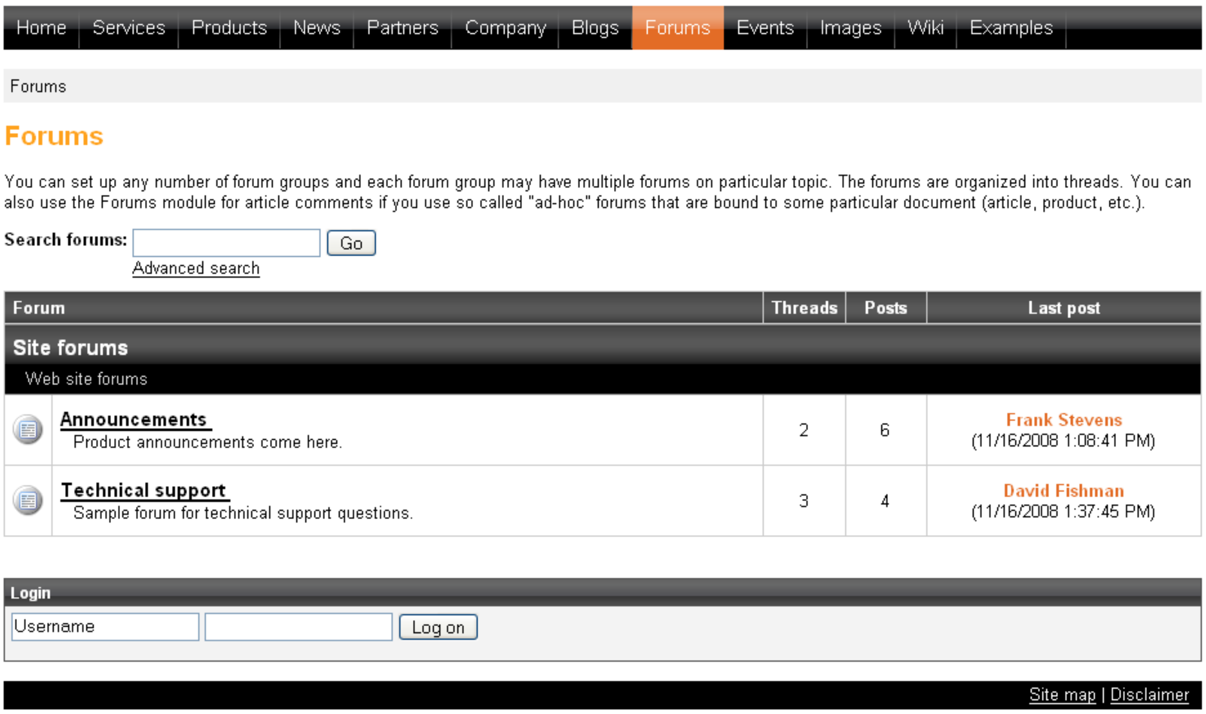
# Part XX

# 40 Module Forums

## 40.1 Forums module overview

This module allows you to integrate full-featured forums into your web site. There are two basic types of forums:

- **Pre-defined forums** - created by the administrator and then displayed on the web site.
- **Ad-hoc forums (article comments)** - created for a single document when the visitor posts the first comment to the given document.




The forums are highly configurable and allow you to:

- Organize forums into forum groups
- Open/close forums
- 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 all posts added to the forum or thread
- Enable forum only for authenticated users
- Specify user roles that are allowed to post to the forum

The Forum module can be managed in **CMS Desk -> Tools -> Forums**.

**General Settings**

You can configure the e-mail address from which all forum notification e-mails are sent in **Site Manager -> Settings -> Forums -> Send forum e-mails from**.

## 40.2 Creating a new forum

The **"pre-defined" forums** need to be created before you publish them on the web site. Each forum must be created within some particular forum group. A **forum group** usually contains forums related to the same topic. For example:

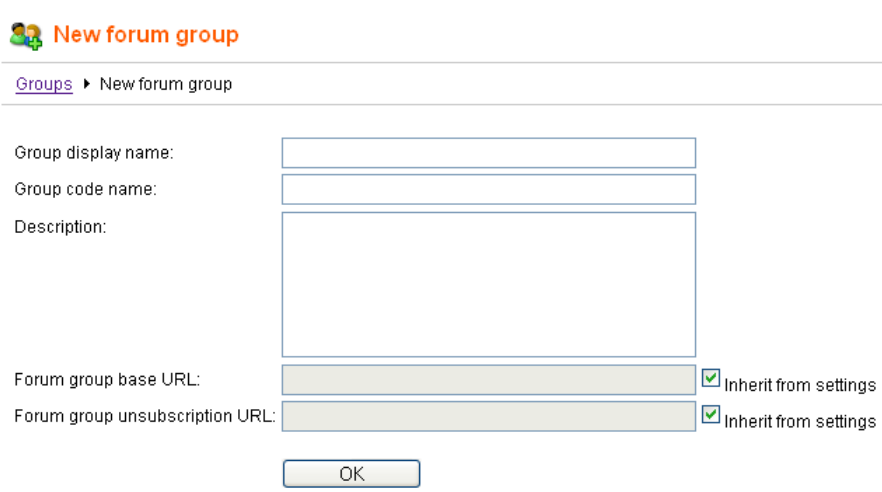
- Computers (forum group)
  - Announcements (forum)
  - Technical questions (forum)
  - FAQ's (forum)
- Web design (forum group)
  - CSS (forum)
  - XHTML (forum)

### Creating a new forum group

Go to **CMS Desk -> Tools -> Forums** and click **Add forum group**. Enter for example:

- **Group display name:** the forum group name displayed on your web site
- **Group code name:** the forum group name used in your code
- **Description:** description of the forum group displayed on your web site
- **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.



## Creating a new forum

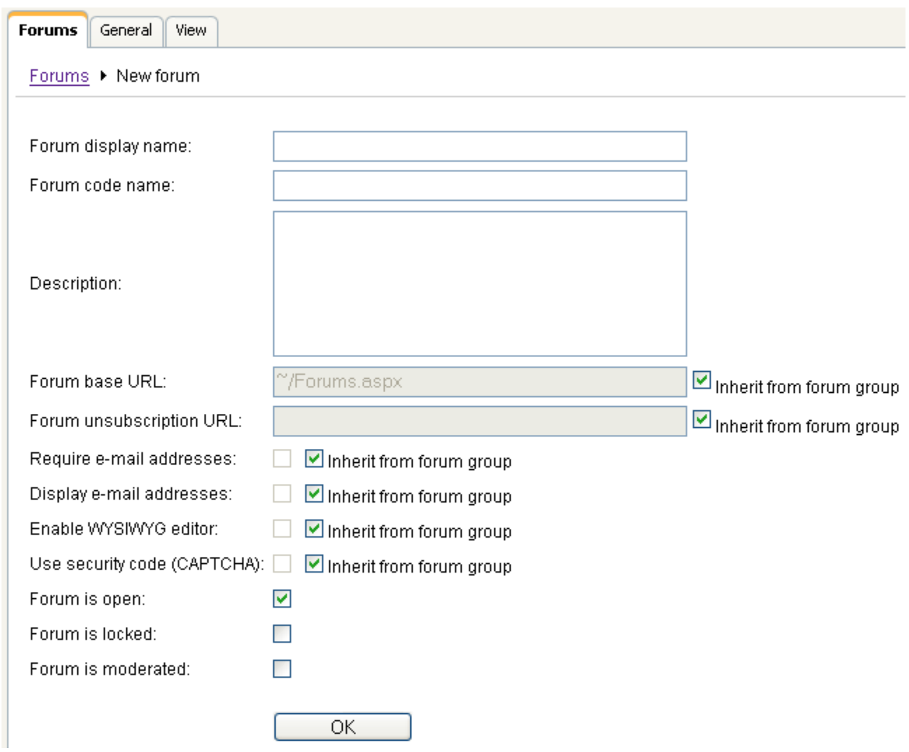
Go to **CMS Desk -> Tools -> Forums** and click **Edit** for some forum group. Go to the Forums tab and click **Add forum**. Enter for example:

- **Forum display name:** the name of the forum displayed on your web site
- **Forum code name:** the name of the forum used in your code
- **Description:** the description displayed on your web site
- **Forum base URL:** URL displayed when the user accesses the forum; *e.g. ~/MyForums.aspx*
- **Forum unsubscription URL:** URL of a 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 forum moderator

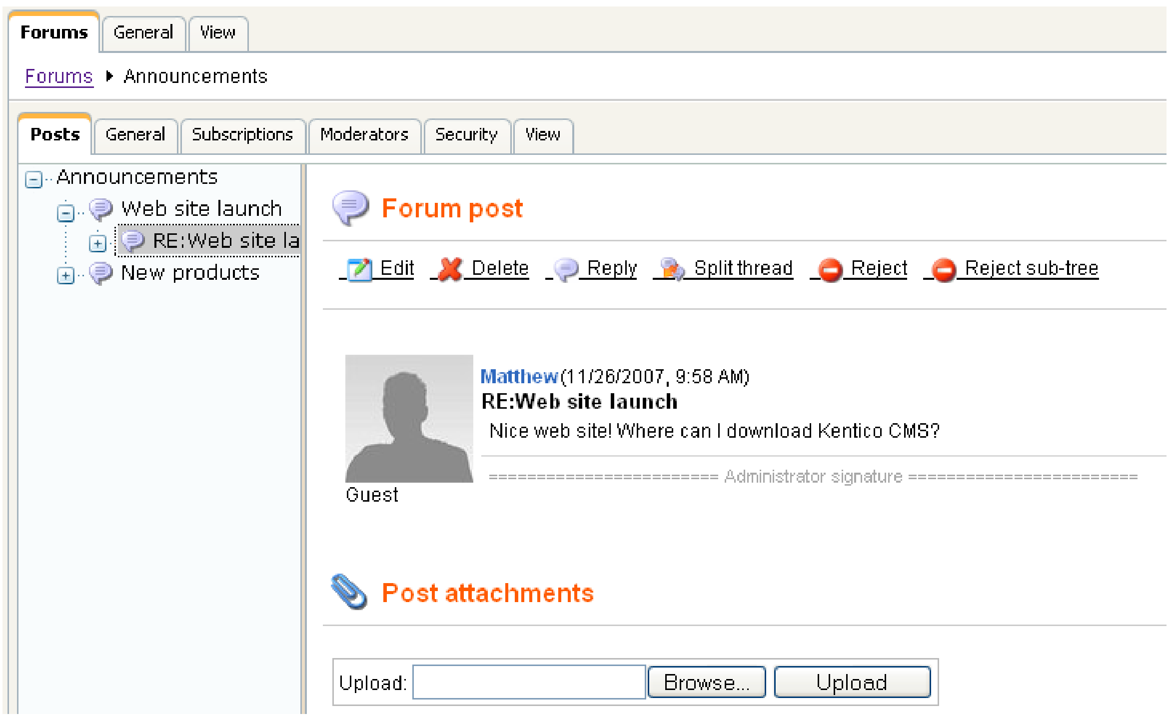
Click **OK** to save.



## 40.3 Managing forum posts

You can manage forum posts directly from the administration interface, in the **Posts** tab in the **Forum properties** dialog. You can perform the following actions with a post selected in the tree on the left:

- **Edit** - you can edit post text, user's name, signature, etc.
- **Delete** - you can delete the post
- **Reply** - you can send a reply to the post
- **Split thread -** you can move the post and its replies to a new thread
- **Reject** - rejects the post so that it is not displayed and needs approval to be displayed
- **Reject subtree -** performs the Reject action for the current posts and all its children



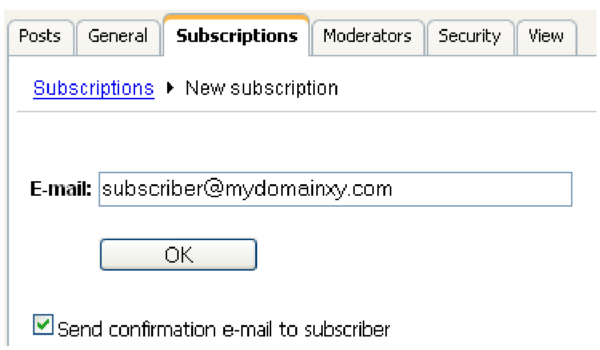
## 40.4 Subscriptions

Subscriptions allow users to receive e-mail notification when a new post is added to the selected forum.

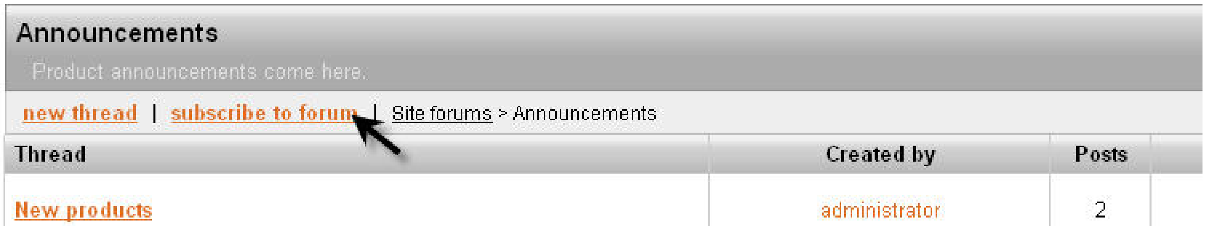
There are several ways how a user can subscribe for receiving e-mail notifications:

1. The forum administrator can subscribe users to the whole forum and manage all subscriptions in Forum Properties -> Subscriptions.

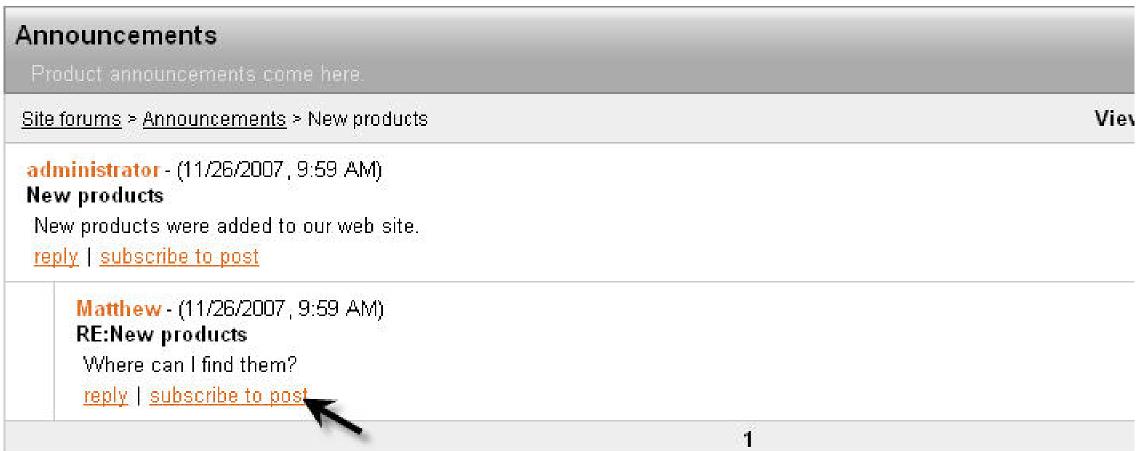

After clicking the **New subscription** link, the following dialog depicted in the screenshot below will be displayed. You only need to enter the user's e-mail address and click **OK**. If you do it with the **Send confirmation e-mail to subscriber** check-box checked, the user will receive a confirmation e-mail, telling her that she has 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 particular post and sub-posts:




The e-mails are sent to the subscribed users based on the **Forum 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**

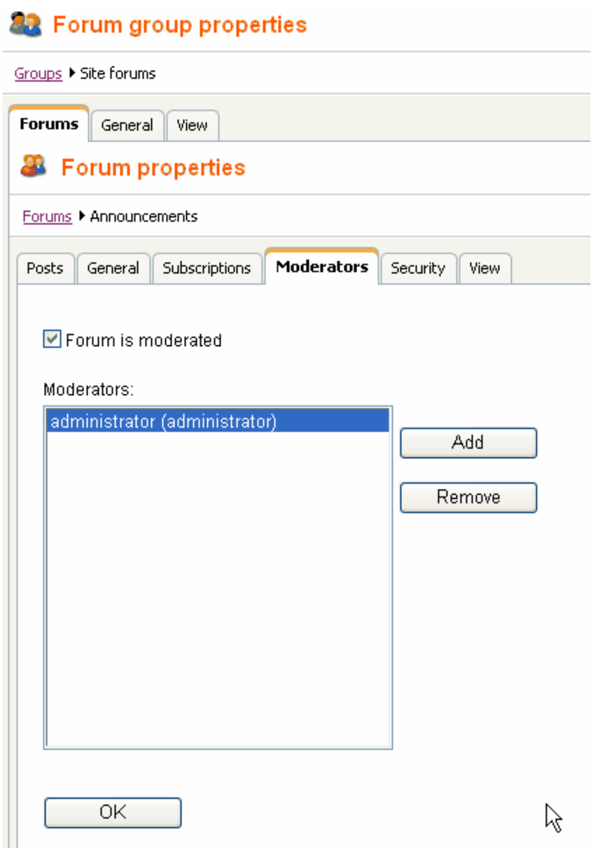
It's important to set a correct **Forum base URL value** in the forum properties so that the notification e-mails contain valid links to the forum. Use URL in format http://localhost/kenticocms/forums.aspx **- enter the full URL of the page with forum, including the domain.**

# 40.5 Forum moderation

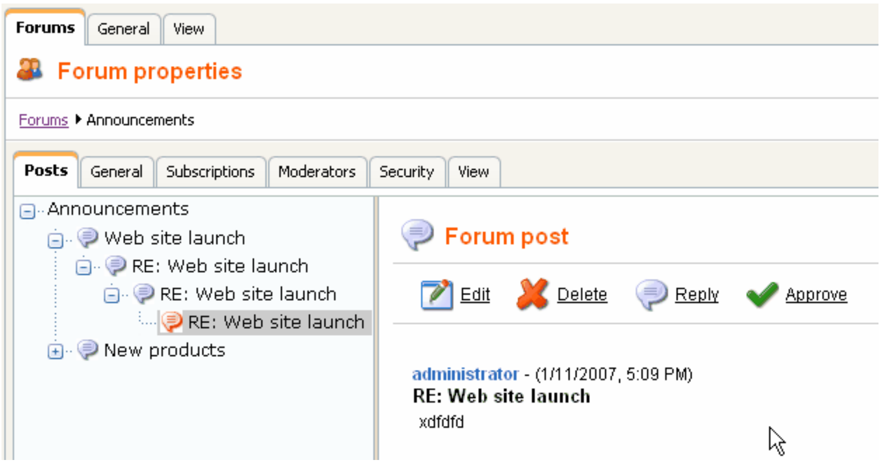
You can configure forum so that it is moderated. It means that all posts must be approved by one of the forum moderators before they are published in the forum.

## Configuring forum moderation

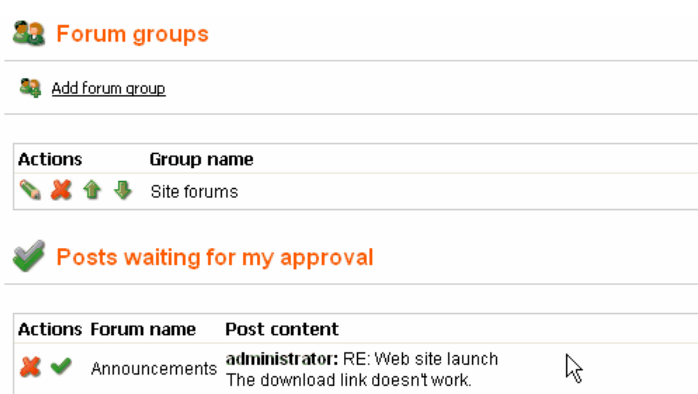
You can enable forum moderation and configure the list of moderators in **Forum properties -> Moderators** dialog. Even when this option is disabled, approval of posts might be required, e.g. when a Bad word is detected or when a post has been rejected.



If you're a moderator, **you can moderate the forum posts either directly on the web site or in the Forum properties -> Posts** section:




... 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**.

# 40.6 Publishing pre-defined forum on the web

When you want to publish a forum on your web site, 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 and Controls 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 forum group on the web site

You can publish the whole forum group on the web site 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:

| Forum | Threads | Posts | Last post |
|---|---|---|---|
| **Site forums**<br>Web site forums | | | |
| **Announcements**<br>Product announcements come here.<br>Lock | 3 | 6 | Frank Stevens<br>(11/16/2008 1:08:41 PM) |
| **Technical support**<br>Sample forum for technical support questions.<br>Lock | 3 | 4 | David Fishman<br>(11/16/2008 1:37:45 PM) |

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**

**Announcements**
Product announcements come here.

New thread | Subscribe to forum | Site forums > Announcements

| Thread | Created by | Posts | Views | Last post |
|---|---|---|---|---|
| Web site launch<br>Lock Unstick thread Move down Move up | administrator | 3 | 259 | administrator<br>(10/19/2008 5:45:18 PM) |
| New products<br>Lock Stick thread | administrator | 3 | 57 | Frank Stevens<br>(11/16/2008 1:08:41 PM) |

1

**2) As a tree of threads and posts**
- set the **Forum layout** property to **Tree**


## Adding forum full-text search

If you want to let users search the forum posts for given text, you can use the **Forum search** (ForumSearch.ascx) and **Forum search results** (ForumSearchResults.ascx) web parts. You will typically place them both on the same page. However, if you need 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`**.


# 40.7 Adding 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", the forum will be displayed on the web site, but it will be actually created only after some visitor adds the first post to the forum. After that, the forum will be created in the **AdHoc forum group** forum group in **CMS Desk -> Tools -> Forums**.

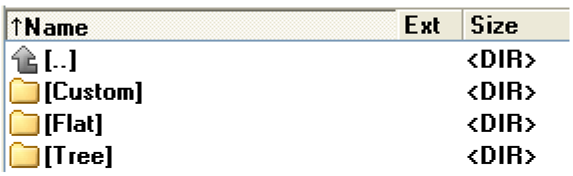
Ad-hoc forums are uniquely identified by the document they belong to.

It is **deprecated** to use the following two web parts, as they are available only because of **backward compatibility** with versions prior to 4.0:

- **Forum (Single forum - Flat)** - this web part ensures backward compatibility with the former **Forum thread list** web part
- **Forum (Single forum - Tree)** - this web part ensures backward compatibility with the former **Forum tree** web part

## 40.8 Customizing forum design

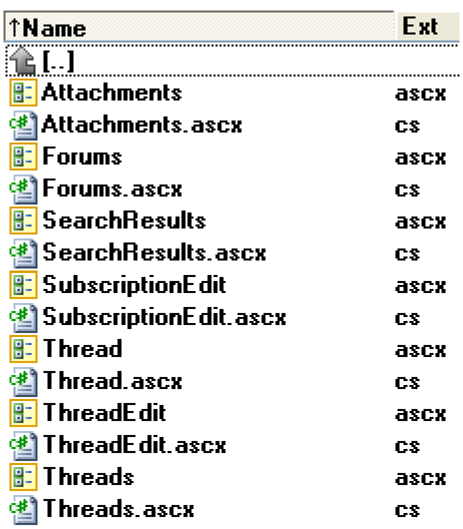
Forum layouts are stored in **<web project>\CMSModules\Forums\Controls\Layouts**. The folder contains three sub-folders by default.



| ↑Name | Ext | Size |
|---|---|---|
| [..] | | <DIR> |
| [Custom] | | <DIR> |
| [Flat] | | <DIR> |
| [Tree] | | <DIR> |

The **Flat** and **Tree** folders contain controls for the alike 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. 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 |

## 40.9 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 chapter for more details. There are also some additional settings related to attachment:

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:

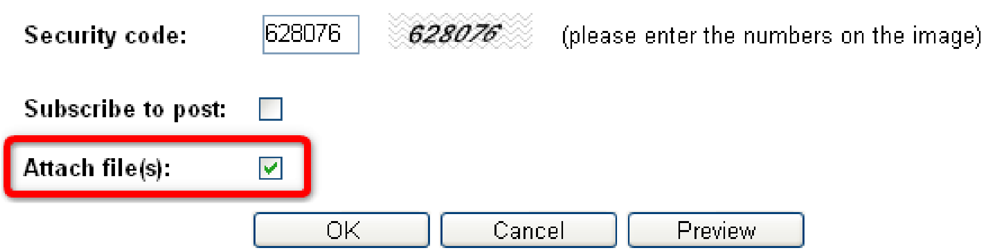
- **Display attachment image** - if checked, attached images will be displayed as images, not only as links
- **Attachment image maximal side size** - if an attached image is larger than the entered value, it will be resized so that its larger side's size is equal to the entered value

When editing a forum in **CMS Desk -> Tools -> Forums**, you can set the following property on the **General** tab:

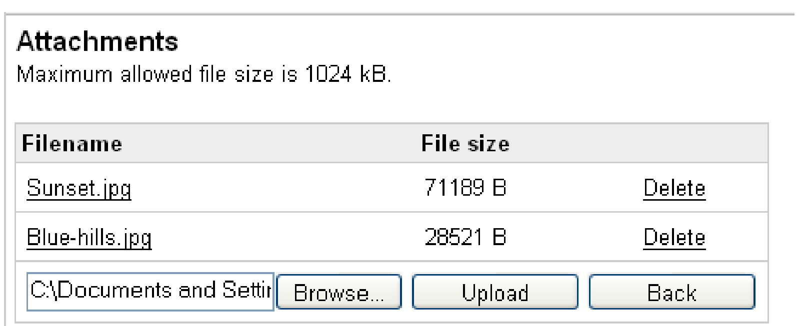
- **Attachment max. file size (kB)** - you can define the maximal size of an attached file in kB

## Adding attachments to a forum post

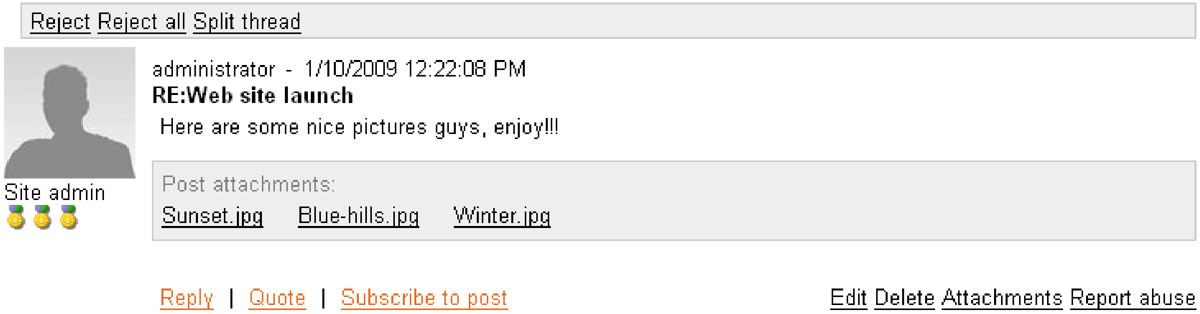
When adding a post to a forum, users can check the **Attach file(s)** check box at the bottom of the post adding dialog.




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.


Reject Reject all Split thread
administrator - 1/10/2009 12:22:08 PM
RE:Web site launch
Here are some nice pictures guys, enjoy!!!
Site admin
Post attachments:
Sunset.jpg Blue-hills.jpg Winter.jpg
Reply | Quote | Subscribe to post
Edit Delete Attachments Report abuse

# 40.10 BBCode support

BBCode is a lighweight markup language designed to let users format their messages. It can be used in many forums on the web. Its tags are similar to HTML tags and are entered in square brackets.

Users can use Bulletin Board code in their forum posts in case that it is enabled. You have to allow use of BBCode in properties of the forum, on the **General** tab. The following table explains particular BBCode tags and properties that have to be enabled on the General tab in order for the tags to be functional.

| Example | Description | Property on General tab |
|---|---|---|
| [url]http://example.org[/url]<br>[url=http://example.com]Example[/url] | Makes the text link leading to the URL. | Enable URL macros in posts |
| [img]http://www.imagesxy.com/img.bmp[/img]<br>[img=200x50]http://www.imagesxy.com/img.bmp[/img]<br>[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>*. | Enable image macros in posts |
| [quote]quoted text[/quote]<br>[quote=Administrator]quoted text[/quote] | Displays text in a grey box; used for quotations.<br>The optional parameter displays<br>*Administrator wrote:*<br>and the quoted text on a new line. | Enable quote macros in posts |
| [code]code example[/code] | Displays text in monospaced format; used for code snippets. | Enable code snippet macros in posts |
| [b]bold text[/b]<br>[strong]bold text[/strong] | Makes the text bold. | Enable bold font macros in posts |
| [i]italicized text[/i]<br>[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]<br>[color=#f00]Red Text[/color]<br>[color=f00]Red Text[/color]<br>[color=#ff0000]Red Text[/color]<br>[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 an 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 the BBcode macros

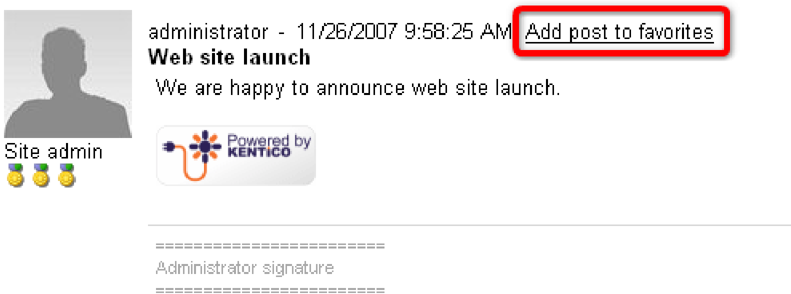
There is an easy way how to resolve macros in ASPX or codebehind code. To resolve all the macros (recommended), use method:

string **CMS.CMSHelper.CMSContext.ResolveDiscussionMacros**(string **inputText**)

Or you can use class **CMS.GlobalHelper.DiscussionMacroHelper** to resolve macros with particular settings using method **ResolveMacros** from the object of this class.

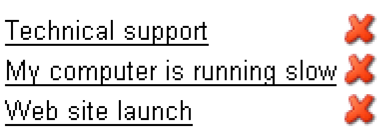
## 40.11 Forum favorites

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 user clicks this link, the post will be added to his 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.



# 40.12 Friendly URLs

Forums display content based on the current values of the **forumid** and **threadid** query string parameters. By default, URL of a forum thread might look like the following:

**<domain>/Forums.aspx?forumid=3&threadid=12**

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}**

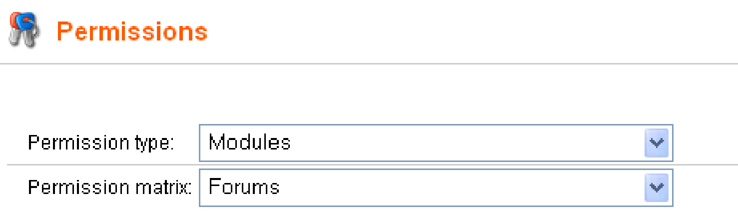
# 40.13 Security

There are two parts of the Forums module's security model:

1. The security of the Forums module administration interface.
2. The security of the forums published on the web site.

## Forums module administration interface

Access to the Forums module administration interface in **CMS Desk -> Tools** can be managed in the **Modules -> Forums** permission matrix in **CMS Desk -> Administration -> Permissions**:




| | Display module | Modify | Read |
|---|---|---|---|
| Authenticated users | | | |
| CMS Community administrators | ✓ | ✓ | ✓ |
| CMS Designers | | | |
| CMS Desk administrators | ✓ | ✓ | ✓ |
| CMS Editors | ✓ | | ✓ |
| CMS Readers | ✓ | | ✓ |
| Everyone | | | |
| Live ID users | | | |
| Non-authenticated users | | | |

- **Display module -** the module will be displayed in the administration interface
- **Modify** - allows users to modify forum settings
- **Read** - allows users to only read forum settings

The 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 web site

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:

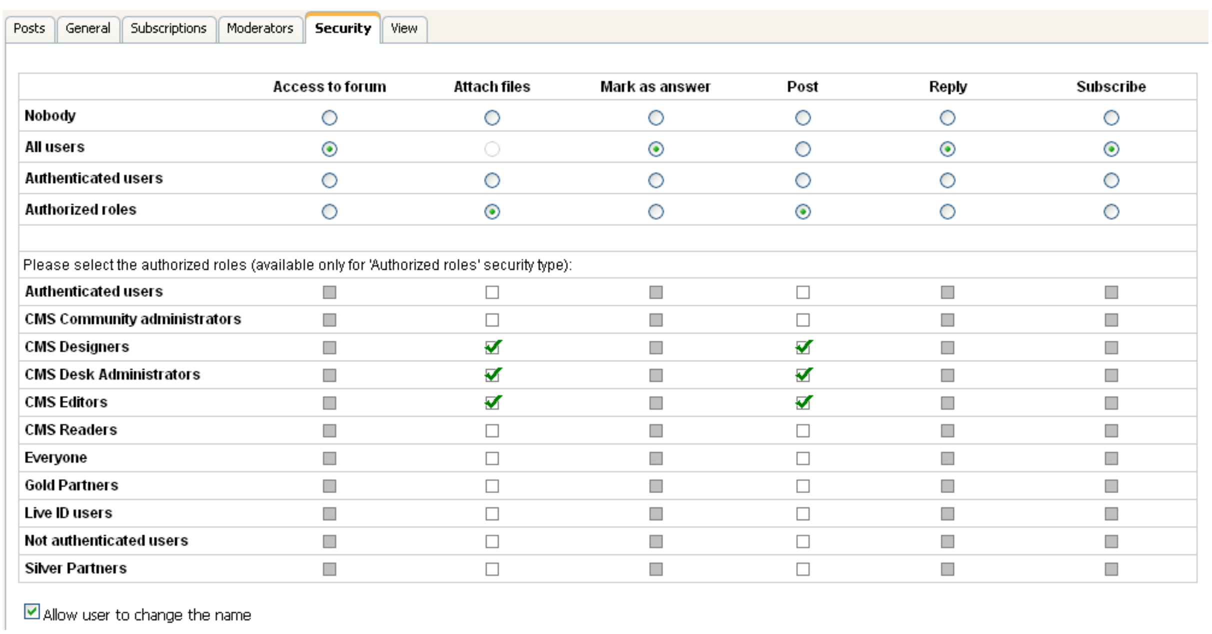
- **Access to forum** - defines who can enter the forum and view posts
- **Attach files** - defines who can attach files to forum posts
- **Mark as answer** - defines who can mark posts as answers in Question - Answer forums
- **Post** - defines who can add posts to the forum
- **Reply** - defines who 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, user name will be used




The following properties of the **Forum group** web part are also related to forum security:

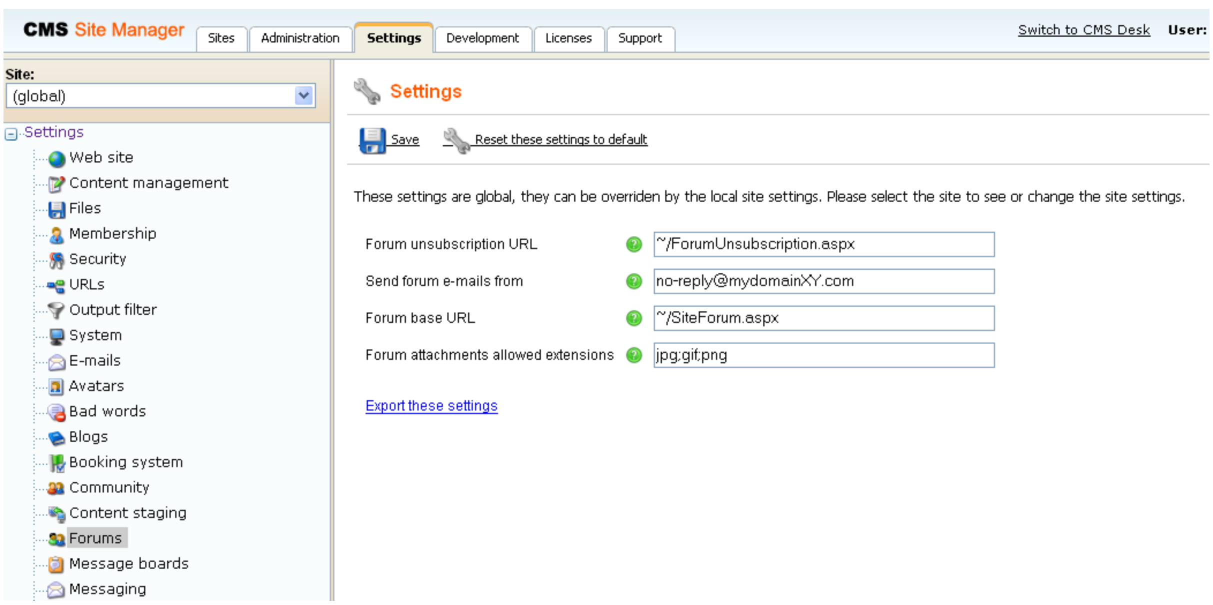
- **Hide forum to unauthorized users** - indicates whether forums for which the user has no permissions are visible for him in the list of forums in a forum group
- **Redirect unauthorized users** - determines whether to redirect unauthorized users to the logon page or whether to display only an info message
- **Access denied page URL** - URL where the user is redirected when trying to access forum for which

she is not authorized

## 40.14 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 should be 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



# Part XX

# 41 Module Friends

## 41.1 Overview

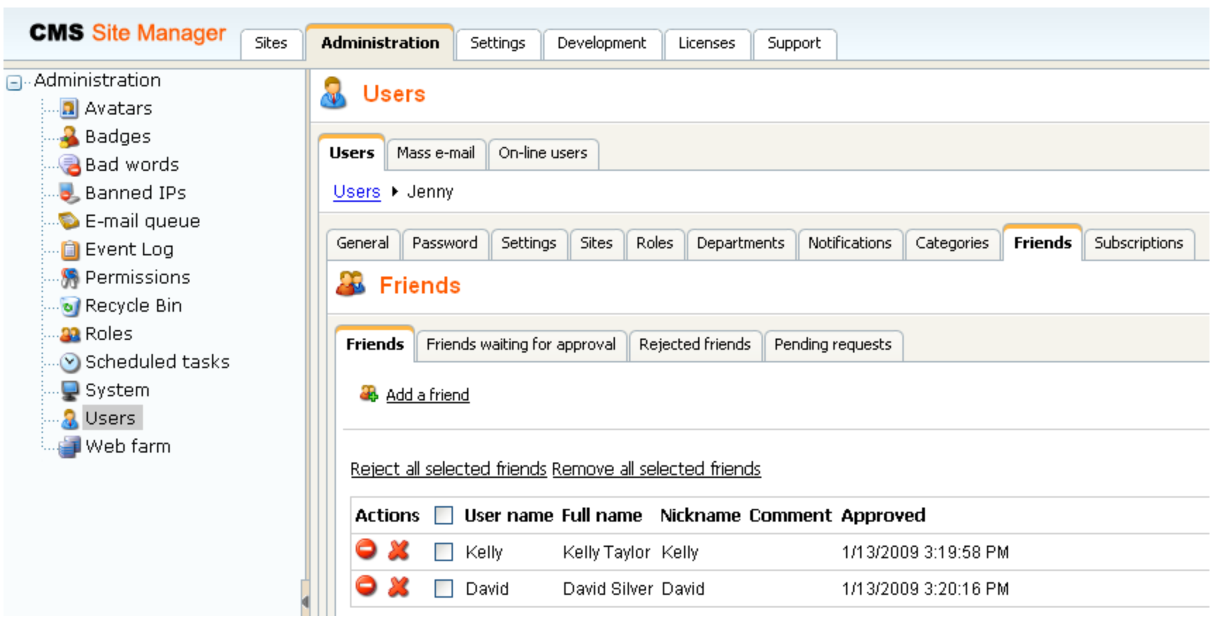
The Friends module allows site members to maintain relationships with other site members and share personal information with them.

## 41.2 Friends management

Site administrators can manage users' friends in both **CMS Desk** and **Site Manager**, in the **Administration -> Users** section. If you choose to **Edit** ( ) some user and switch to the **Friends** tab, you will be offered with the four tabs for friends management that are described below. The same tabs will be offered in **CMS Desk -> My desk -> My friends**, where the currently logged-in user can manage her own friendships. Names of the tabs in **My desk -> My Friends** are stated in brackets:

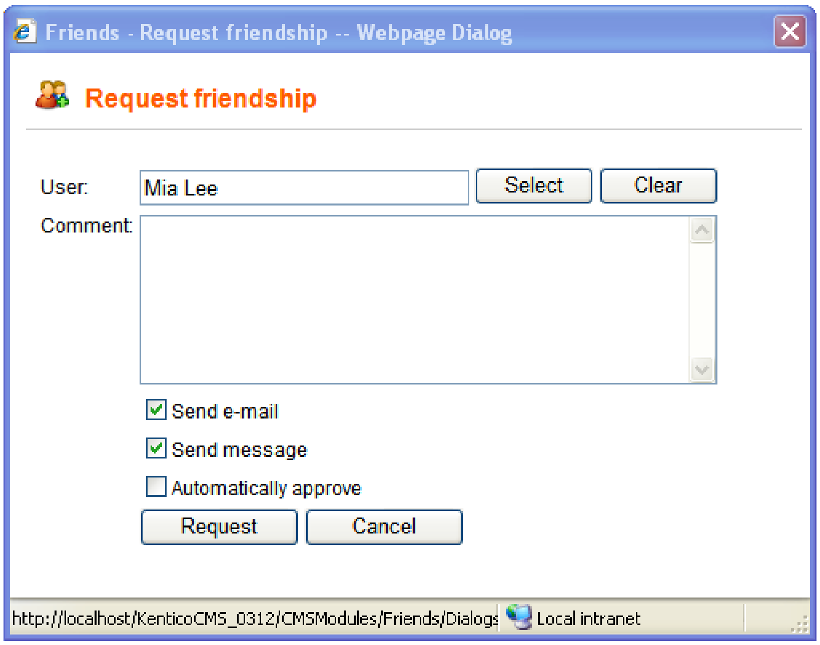
### Friends (My friends)

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. The **Request friendship** link can be used for requesting some user's friendship.



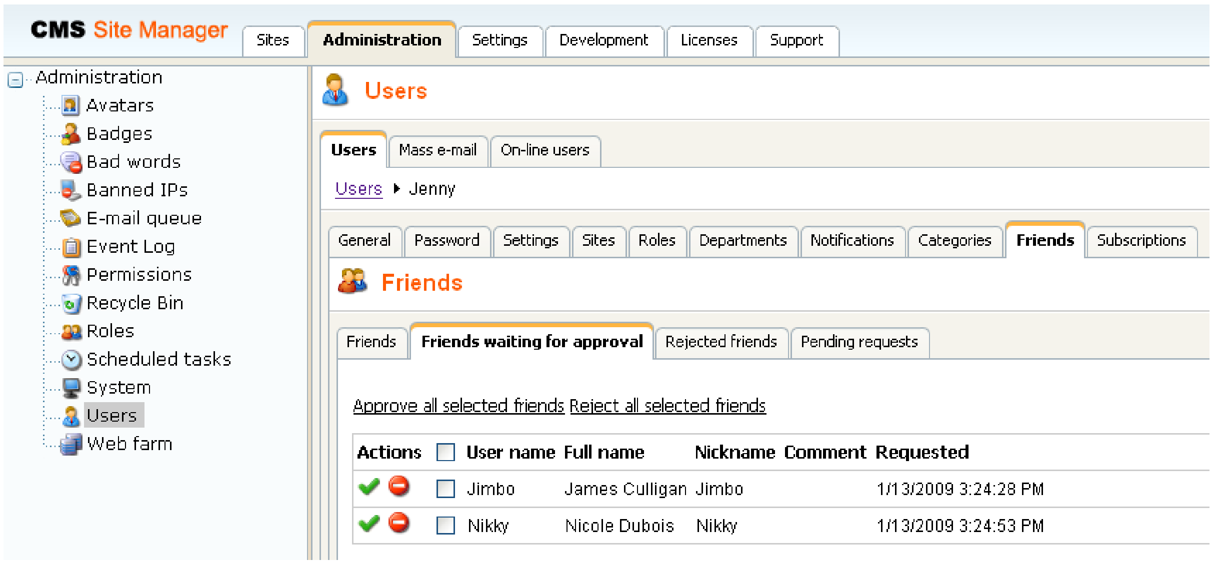
In the Request friendship dialog, you have to click the **Select** button, which opens the familiar user selection dialog. After selecting a user, you can enter some comment that will be displayed to the user when she receives the friendship request. You can also check the following check-boxes:

- **Send e-mail** - an e-mail with the friendship request will be sent to the user
- **Send message** - messaging module message with the friendship request will be sent to the user
- **Automatically approve -** global administrators and users with Manage permission for the Friends module can use this check-box, which creates the requested friendship without the other user's approval



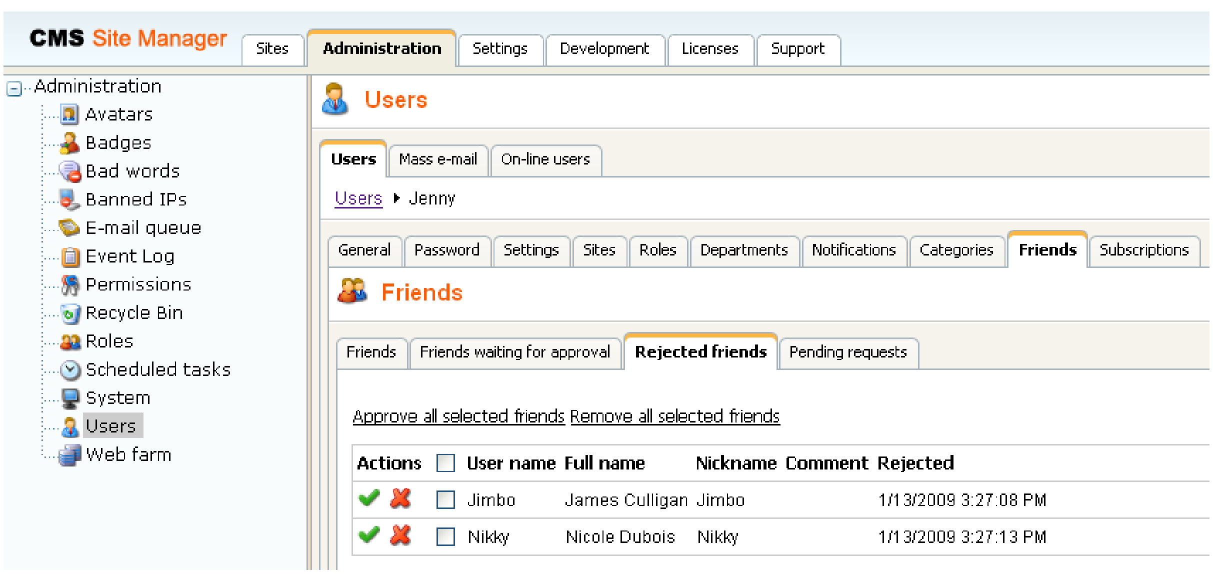
### Friends waiting for approval (Friends waiting for my approval)

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.



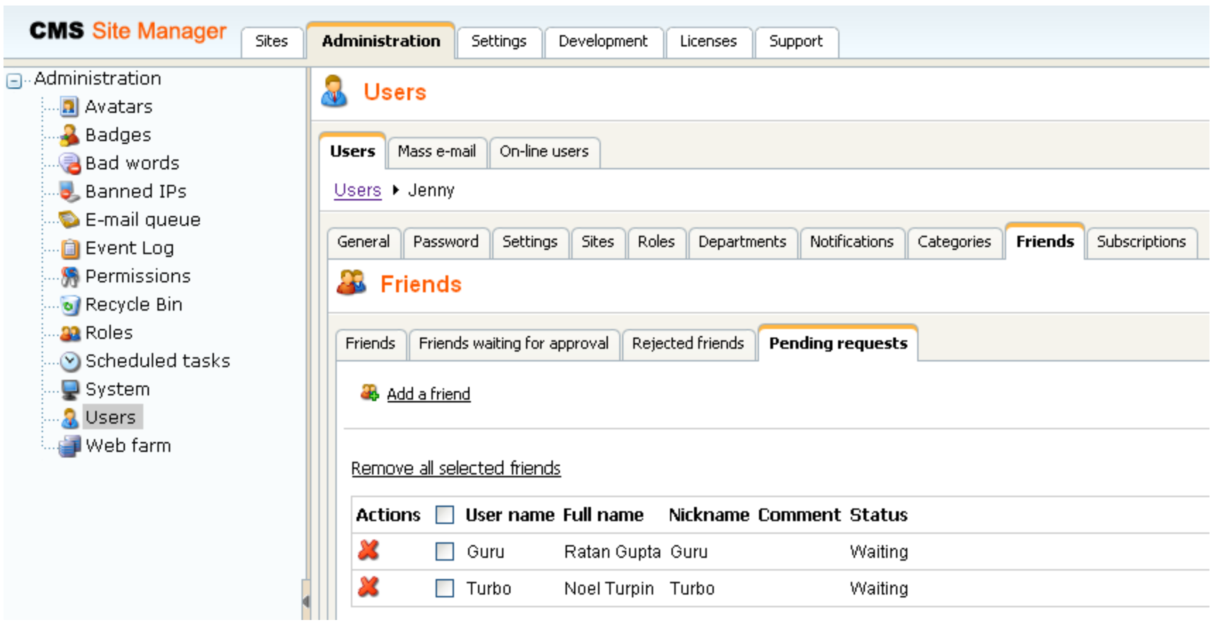

### Rejected friends

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 her your friend, or **Remove** () the user from the list of rejected, which enables her 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.




## 41.3 Available e-mail templates

In **Site manager -> Development -> E-mail templates**, you can find the following e-mail templates:

**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:

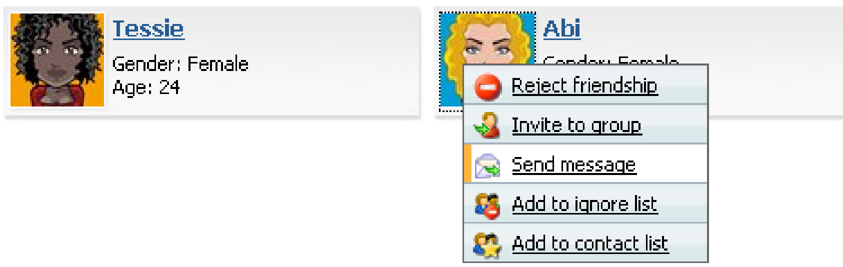
**Text** - text of the template used for e-mails
**Plain text** - text of the template used for messaging module messages

## 41.4 Available web parts

The Friends module comes with the following web parts. For the detailed description of each web part's properties please see **Kentico CMS Web parts** reference that is a part of your Kentico CMS installation and can be accessed through the Kentico CMS folder in Windows' **Start -> All programs** menu.

### Friends viewer

This web part can typically be used on user's profile page for viewing her 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 on-site equivalents of the friends management sections of Kentico CMS administration interface described in chapter Friends management:

### 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 |
|---|---|---|---|---|---|---|
| | ☐ | Tessie | Iman Teshome | Tessie | | 12/3/2008 2:10:23 PM |
| | ☐ | Abi | Abigail Woodwarth | Abi | | 12/3/2008 2:27:56 PM |

### 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 |
|---|---|---|---|---|---|---|
| | ☐ | Josh | Joshua O'Neil | Josh | | 12/3/2008 5:00:40 PM |
| | ☐ | Pogo | Wayne Pronger | Pogo | | 12/3/2008 5:01:21 PM |

### 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 |
|---|---|---|---|---|---|---|
| | | Steevie | Jack Stevenson | Steevie | | 12/3/2008 3:06:54 PM |
| | | Turbo | Noel Turpin | Turbo | | 12/3/2008 3:06:59 PM |

### My pending requests

This web part displays a list of all users whose friendship the current user requested.

Remove all selected friends

| Actions | | User Name | Full Name | Nickname | Comment | Status |
|---|---|---|---|---|---|---|
| | | Nikky | Nicole Dubois | Nikky | | Waiting |
| | | Jenny | Jane Oakley | Jenny | | Waiting |

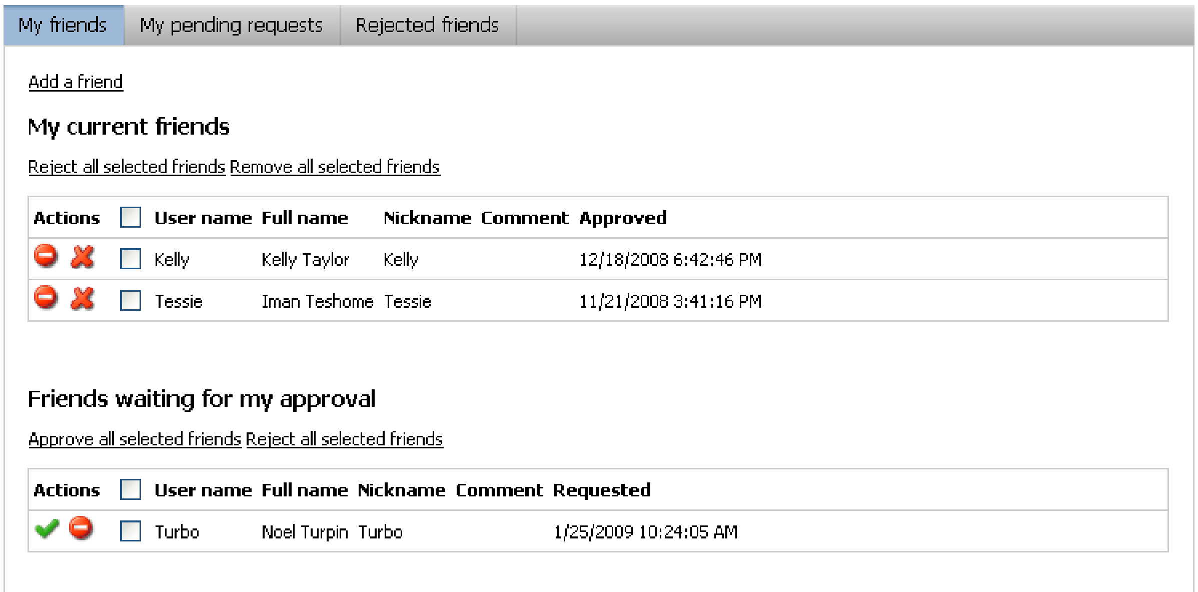
### Request friendship

This web part displays only a link that, when clicked, opens the friendship request dialog.

Request friendship

## My friends

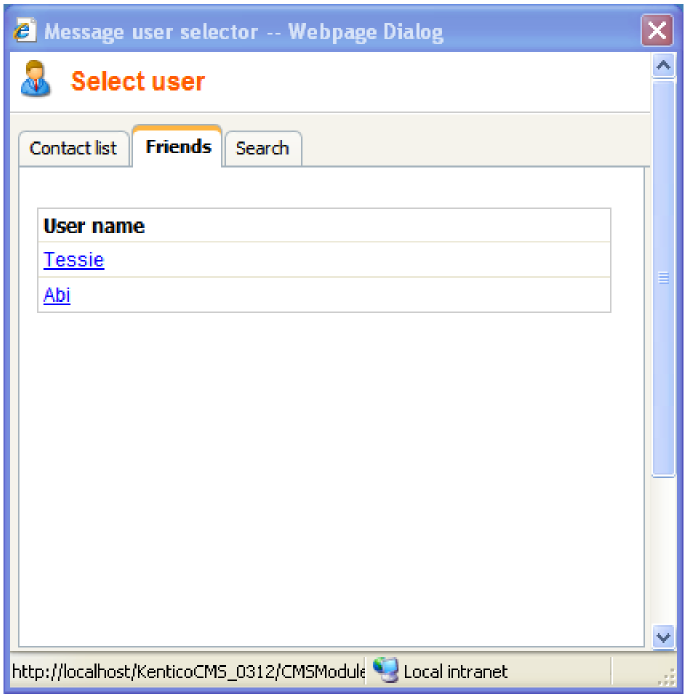
This web part combines the five previously mentioned web parts in one web part. It's functionality is fully equal to the **CMS Desk -> My Desk -> My friends** section of Kentico CMS administration interface.



# 41.5 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 her user name.




## Fields visibility

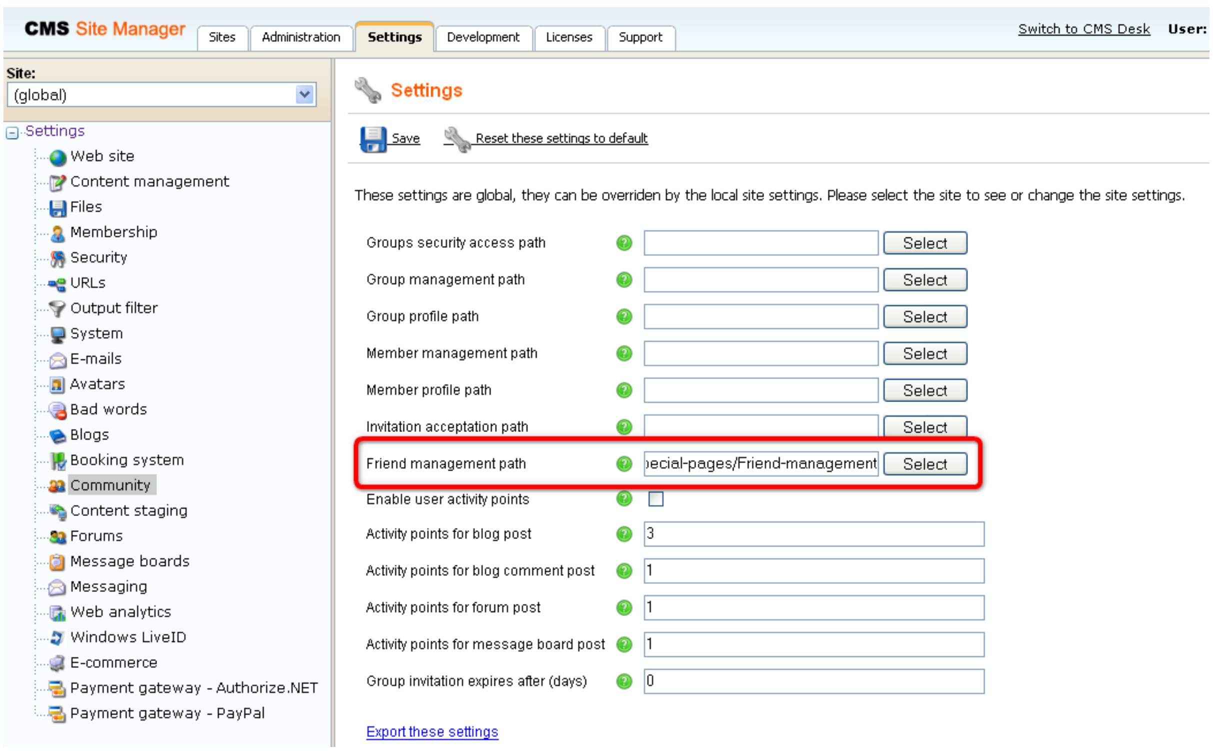
Users can set the visibility of some fields on their public profile to **Display to friends**, which makes the field visible only to their friends. For more information on this topic, please refer to the Custom fields visibility chapter.

## 41.6 Settings

There is only one setting related to the Friends module - **Friends management path**. It is located in **Site Manager -> Settings -> Community**.

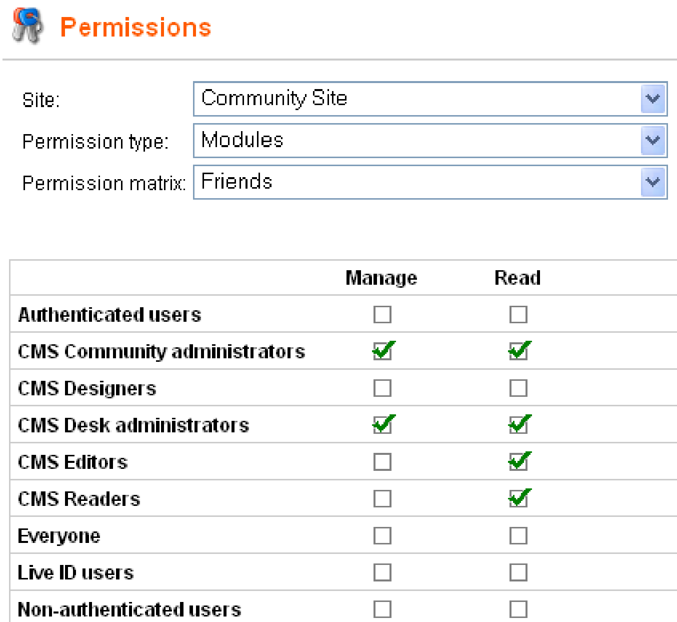
Its value determines the node alias path of the page on that the **Friendship management** web part is placed. This is a special web part that handles friendship management actions from friendship request e-mails.

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, the she is 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.



## 41.7 Security

Permissions of the **Friends** module can be set in both **Site manager** and **CMS Desk**, in the **Administration -> Permissions** section, in 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

# Part XX

# 42 Module GeoMapping

## 42.1 GeoMapping overview

The GeoMapping module allows you to display content on a map. It uses the Google Maps, which is a free service provided by Google available at http://www.google.com/apis/maps/gallery/mapsAPIProducts.html.

This module can be used for many scenarios when you need to display a map:
- show your offices
- show your store
- 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.

### Google Maps Key

The Google Maps system requires a unique key for every domain where you display the maps. Kentico CMS contains keys for localhost and 127.0.0.1. When you move your web site to a real domain, you need to get your own key at http://code.google.com/apis/maps/signup.html and set the **Google maps key** value in the **Google Maps** web part properties.

### How to get a longitude and latitude for a given place

You can use the on-line service that allows you to enter the country, city and street and it shows you to the longitude and latitude.http://world.maporama.com

## 42.2 Example: Displaying offices on the map

This example will show you how to display a list of offices and their location on the map.

### Step 1 - Creating a new page with offices

Sign in as administrator to **Kentico CMS Desk -> Content** and click the root. Click **New** and choose to create a new **Page**. Enter the page name **Offices** and choose to create a **blank page** using the **Simple** layout. Click **Save**.

Now we need to add a list of offices. Switch to the design mode and add the **Repeater** web part. Set the following properties:

- Path: /{0}/%
- Document types: cms.office
- Transformation: cms.office.simple
- Item separator: <br />

Click **OK**.

## Step 2 - Geocoding your information

Now we will create two documents of type **Office.** Click the **Offices** page and click **New**. Choose to create a new **Office** and enter the following values:

**Office 1:**
- Office name: Northwest Transport - New York
- Address line 1: 1290 Avenue of the Americas
- City: New York
- ZIP code: 10104
- State: NY
- Country: USA
- Phone: 123456789
- E-mail: ny@north.com
- Latitude: 40.76
- Longitude: -73.98

Click **Save and another**.

**Office 2:**
- Office name: Northwest Transport - San Francisco
- Address line 1: 835 Market Street
- City: San Francisco
- ZIP code: 94103
- State: CA
- Country: USA
- Phone: 123456789
- E-mail: sf@north.com
- Latitude: 37.78
- Longitude: -122.41

Click **Save**.

**Please note:** The Office document type already contains the **Latitude and Longitude fields**. If you're using a custom document type, you will need to define them. They must be of type **Decimal number**. You can call them as you need.

## 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** mode and add the **Google services/Google Maps** web part.

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:

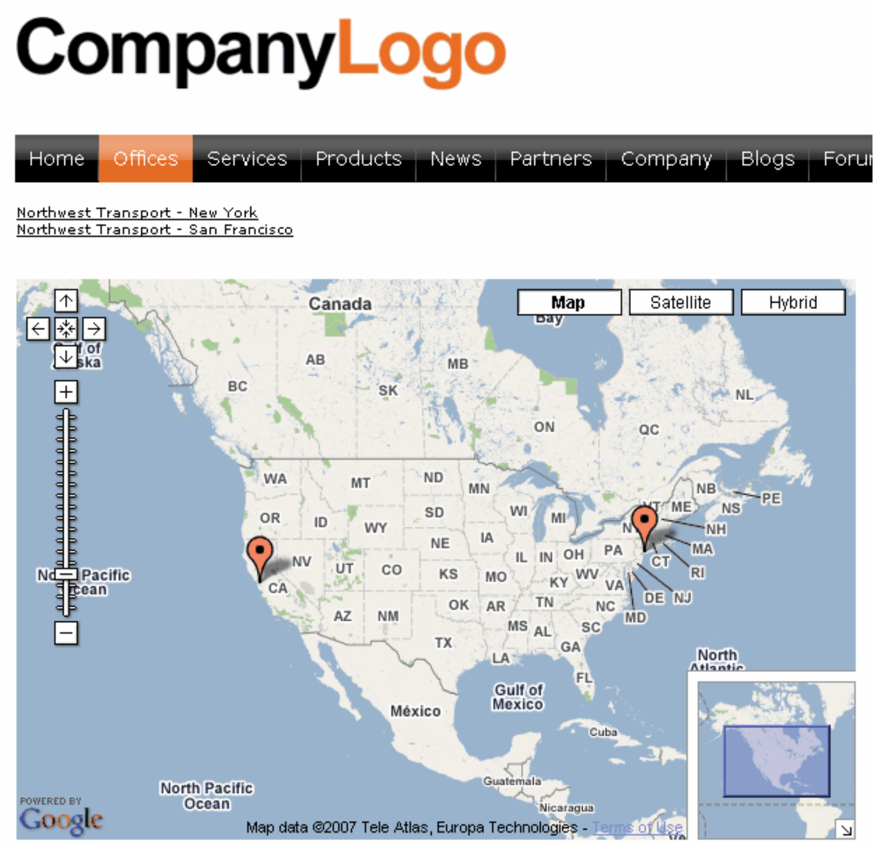
- **Transformation:** cms.office.preview

Now we set the following values that specify how the map is displayed:

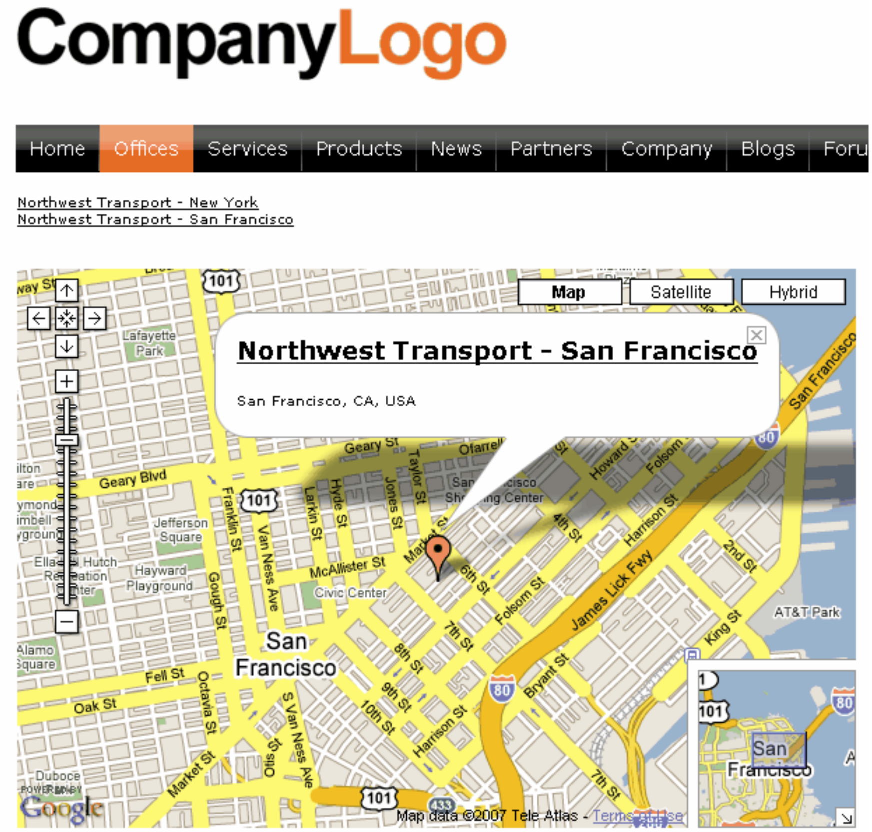
- **Google maps key:** leave this value empty if you're running the web site at http://localhost or http://127.0.0.1. If you use some particular domain name, you need to get a key at http://code.google.com/apis/maps/signup.html.
- **Large view scale:** 7 (the default zoom)
- **Detailed view scale:** 10 (the zoom used when viewing the selected document)
- **Width:** 600 (in pixels)
- **Height:** 400 (in pixels)
- **Default latitude:** 39.27 (latitude of the map center when the overview map is displayed)
- **Default longitude:** -98.20 (longitude of the map center when the overview map is displayed)
- **Latitude field:** OfficeLatitude (the field containing the latitude information of the document)
- **Longitude field:** OfficeLongitude (the field containing the longitude information of the document)
- **Tooltip field:** OfficeName (the field displayed when you mouse-over a balloon representing the object on the map.

Click **OK**.

Sign out and see the page. It will look like this:



The is the overview map with balloons. 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:



# Part XXX

# 43 Module Groups

## 43.1 Overview

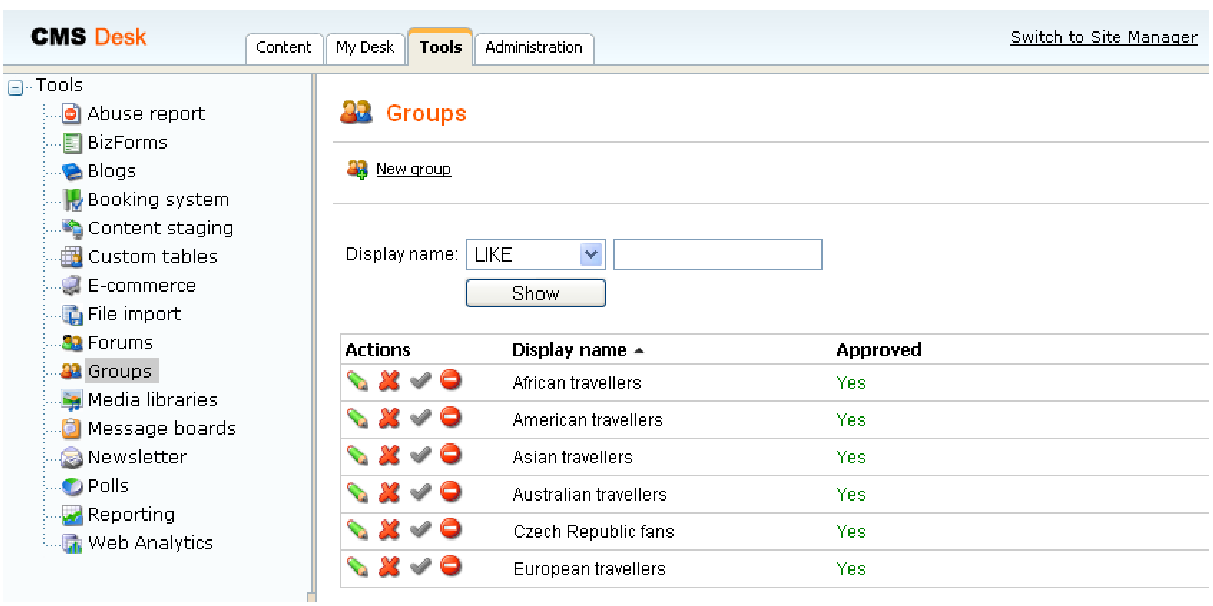
The groups module allows site users to create new groups or join existing ones. Groups are usually related to some topic or field of interest and can have their own documents' section, forums, message boards, media libraries and polls.

## 43.2 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 groups' **display name**.

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 by clicking the **New group** link at the top part of the page. Groups in the list can be **Edited** ( ) or **Deleted** ( ).

In case that 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.



## 43.3 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.

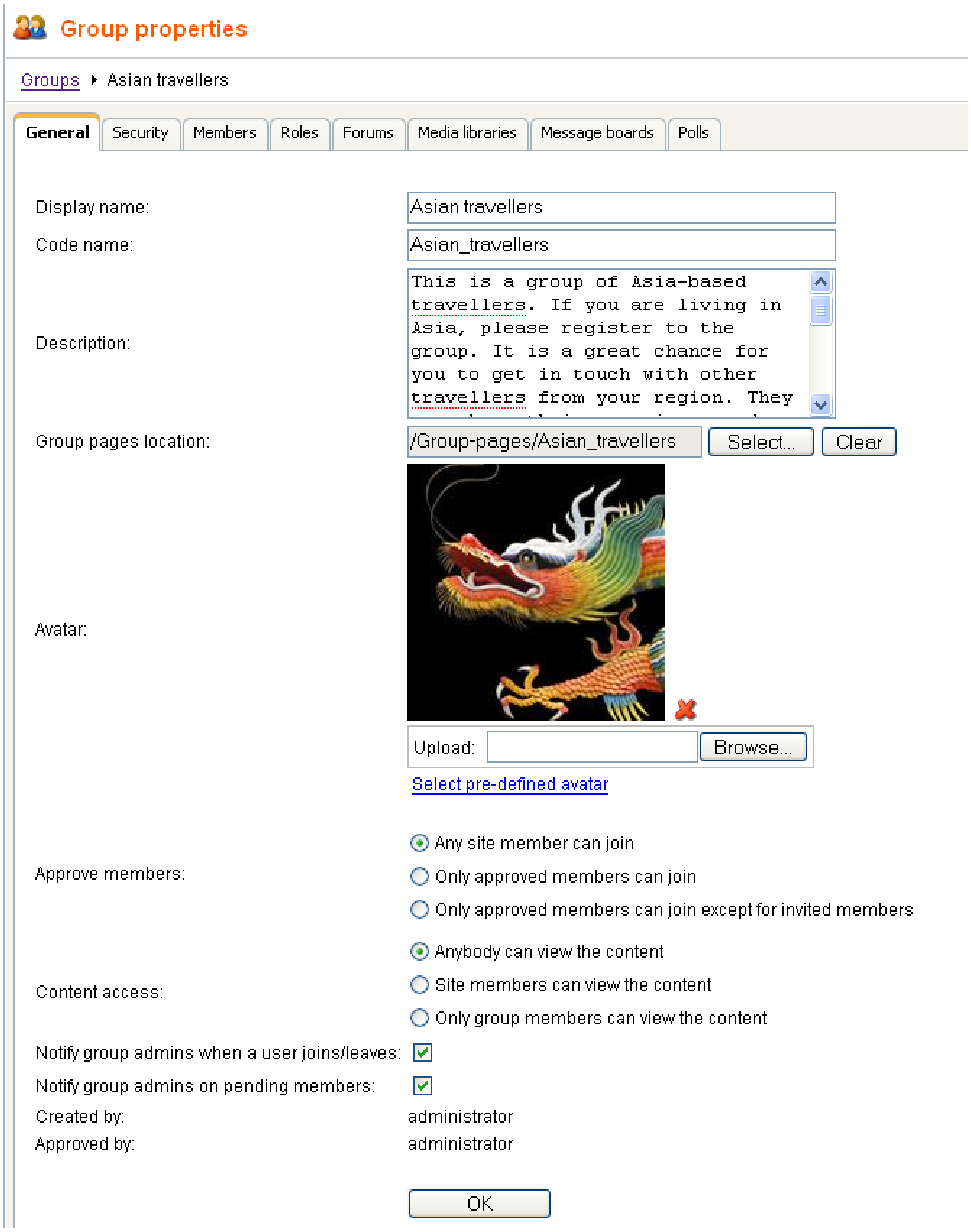
### General tab

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
- **Avatar** - group avatar image

- **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
- **Notify group admins when a user joins/leaves** - if checked, group administrators will receive notification e-mail when a user joins/leaves a group
- **Notify group admins on pending members** - if checked, group administrators will receive notification e-mail when a user requests joining a 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


Group properties
Groups ▸ Asian travellers
General
Security
Members
Roles
Forums
Media libraries
Message boards
Polls
Display name:
Asian travellers
Code name:
Asian_travellers
Description:
This is a group of Asia-based travellers. If you are living in Asia, please register to the group. It is a great chance for you to get in touch with other travellers from your region. They
Group pages location:
/Group-pages/Asian_travellers
Select...
Clear
Avatar:
Upload:
Browse...
Select pre-defined avatar
Approve members:
Any site member can join
Only approved members can join
Only approved members can join except for invited members
Content access:
Anybody can view the content
Site members can view the content
Only group members can view the content
Notify group admins when a user joins/leaves:
Notify group admins on pending members:
Created by:
administrator
Approved by:
administrator
OK

## Security tab

On this tab, you can use the matrix to set permissions for group pages. The following permissions can be assigned:

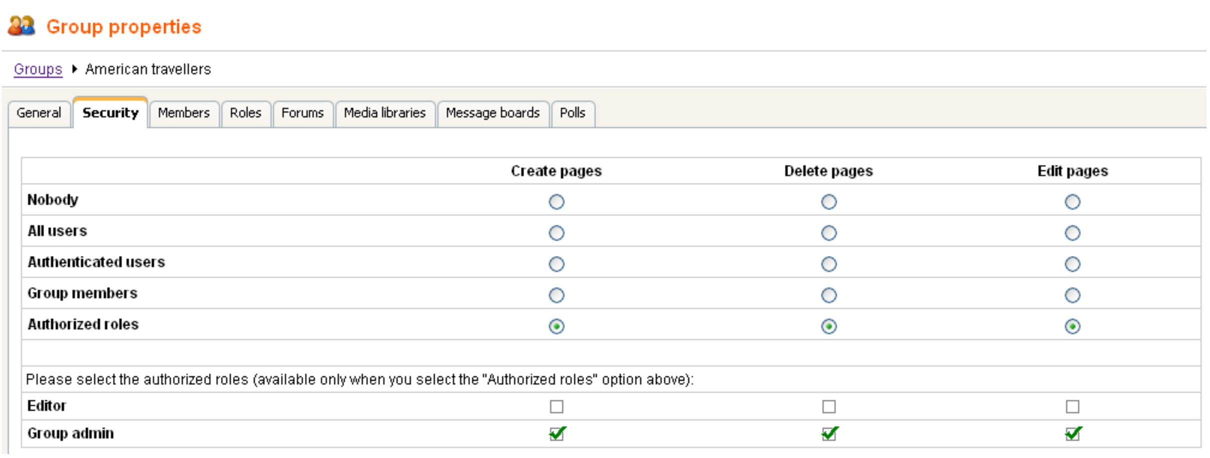
- **Create pages** - users can create group pages
- **Delete pages** - users can delete group pages
- **Edit pages** - users can edit group pages

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 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

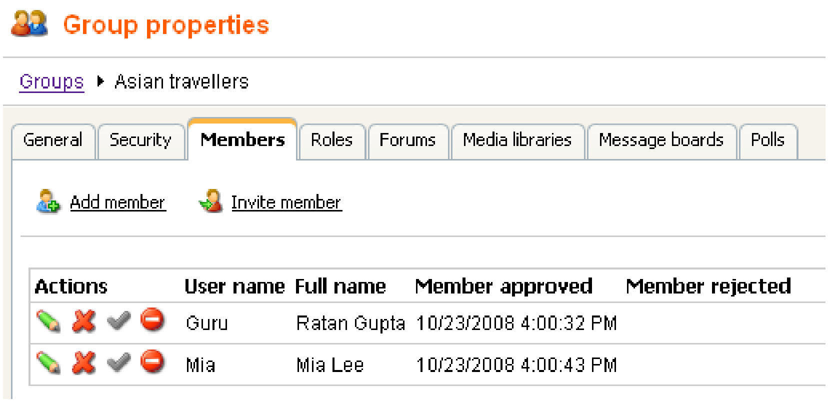
**Group admin's permissions**

Group administrators can perform any of these actions, even if they haven't the permissions assigned.



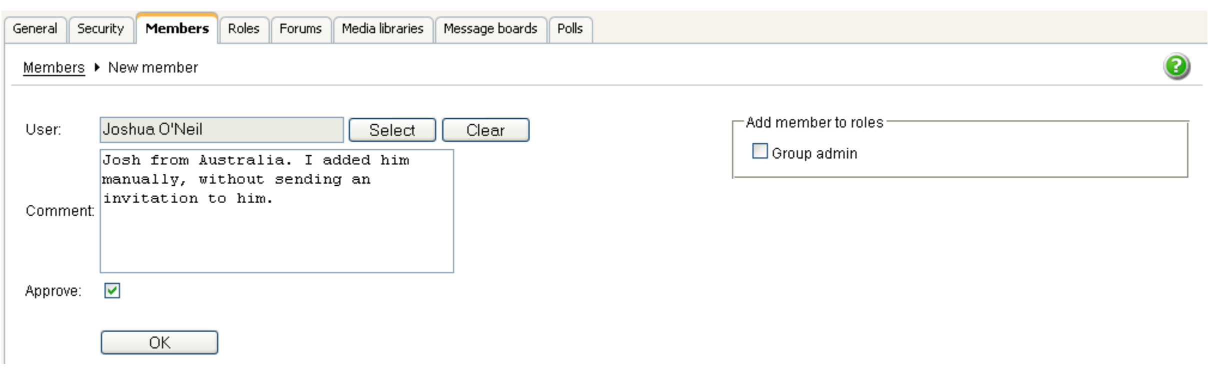
## Members tab

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 can not request joining the group until she is 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 user to a group, you have the following options:

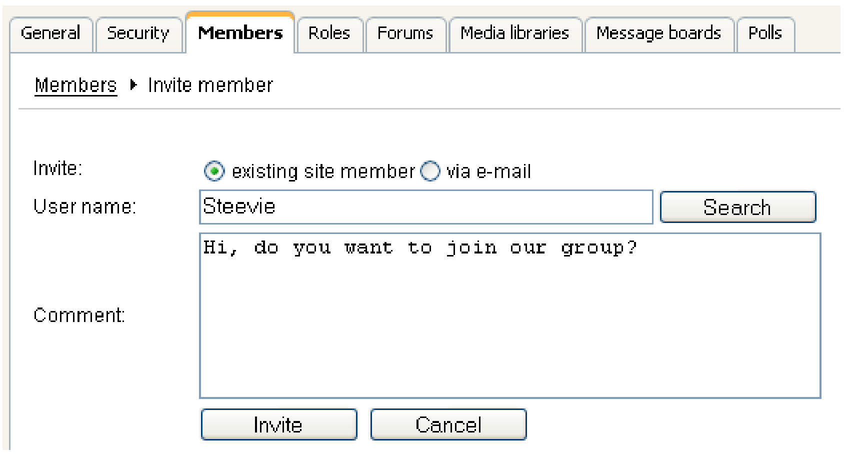
**User** - select an existing site user who you want to add to the group
**Comment** - text comment that you can add to the user; this comment is not sent to the user, it is only displayed in the administration interface
**Approve** - if checked, the user will be automatically approved; if not, user will need group admin's approval before she becomes member of the group
**Add member to roles** - using the check-boxes, you can assign the user to group roles



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



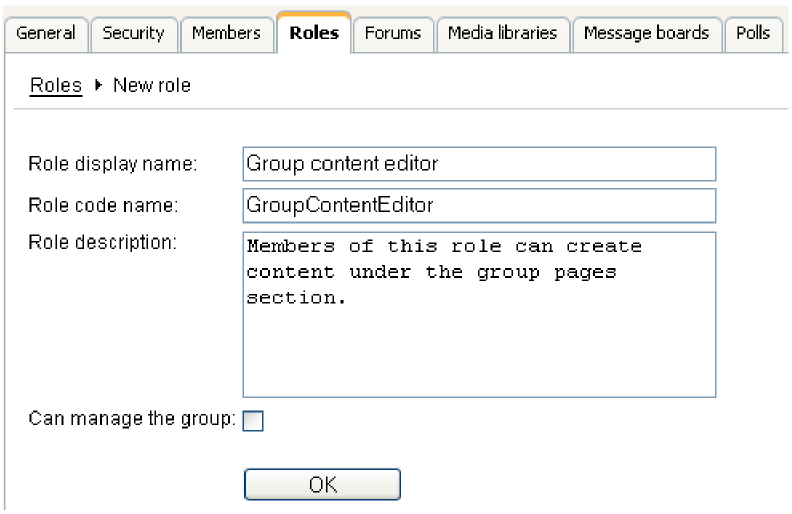

## Roles tab

On this tab, you can see a list roles defined for the group. These roles are applicable only in the context of the group. Don't confuse them with web site roles, which can be set in Site Manager -> Administration -> Roles.

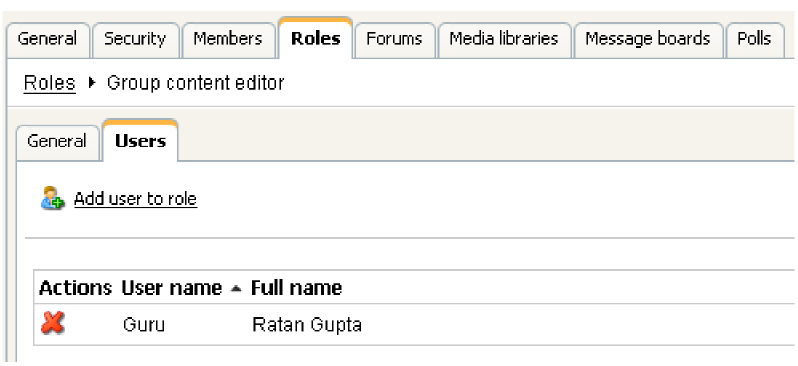
Roles in the list can be **Edited** ( ) and **Deleted** ( ).

If you click the **New role** link above the list, you can define a new role for the group. The following properties need to 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 group, two tabs are offered. On the **General** tab, you can change the details entered when creating the group, as described above. On the **Users** tab, you can see a list of all members of the role. These can be **Deleted** (), which removes them from the role. New users can be added to the role using the **Add user to role** link above the list.



### 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 Module 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 Module 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 Module 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 Module Polls chapter of this guide for more information on their management.

## 43.4 Enabling users to create groups

You can enable site users to create new groups by placing the **Community -> Group registration** web part to your site. You have to set the following properties of the web part:

- **Template source alias path** - alias path of a location where group template source files are stored
- **Template target alias path** - alias path where files created from group template source files will be loaded when the group is created
- **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 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 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

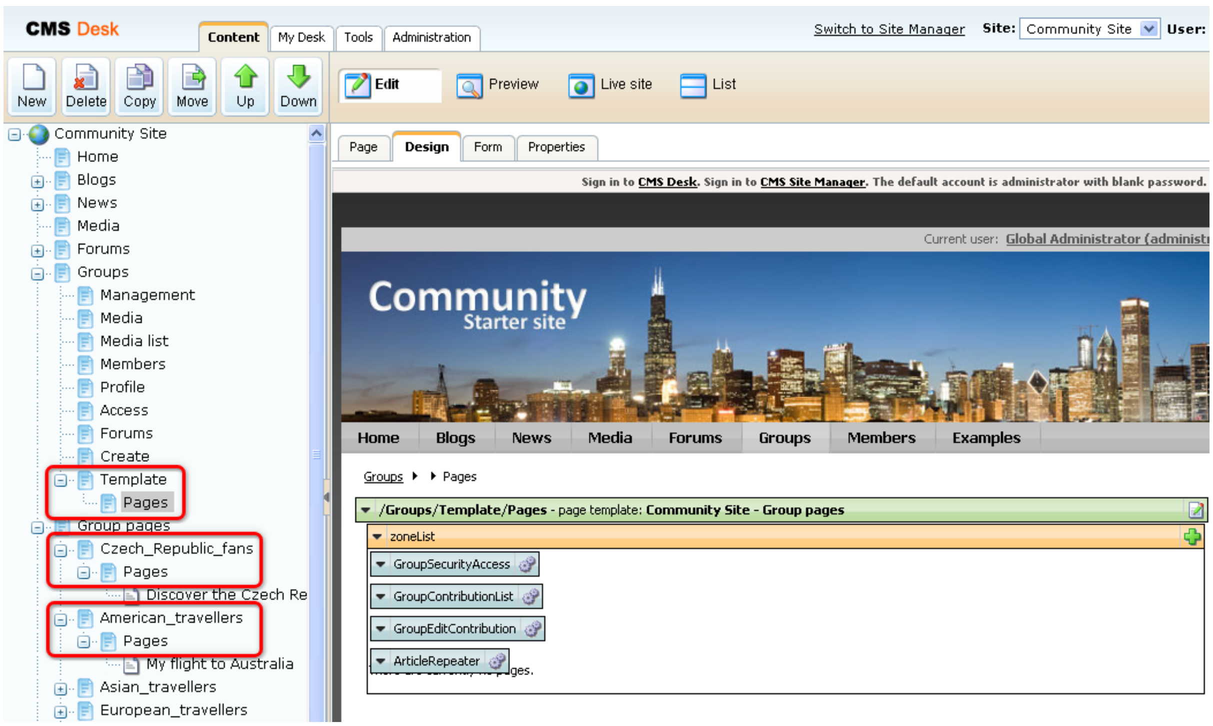
Each group has its own section on the web site where its content is stored - so called group pages. When adding the **Group registration** web part to your site, you have to specify the **Template source alias path** and **Template target alias path** properties. These two properties are essential when 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, it's 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's documents will be stored.

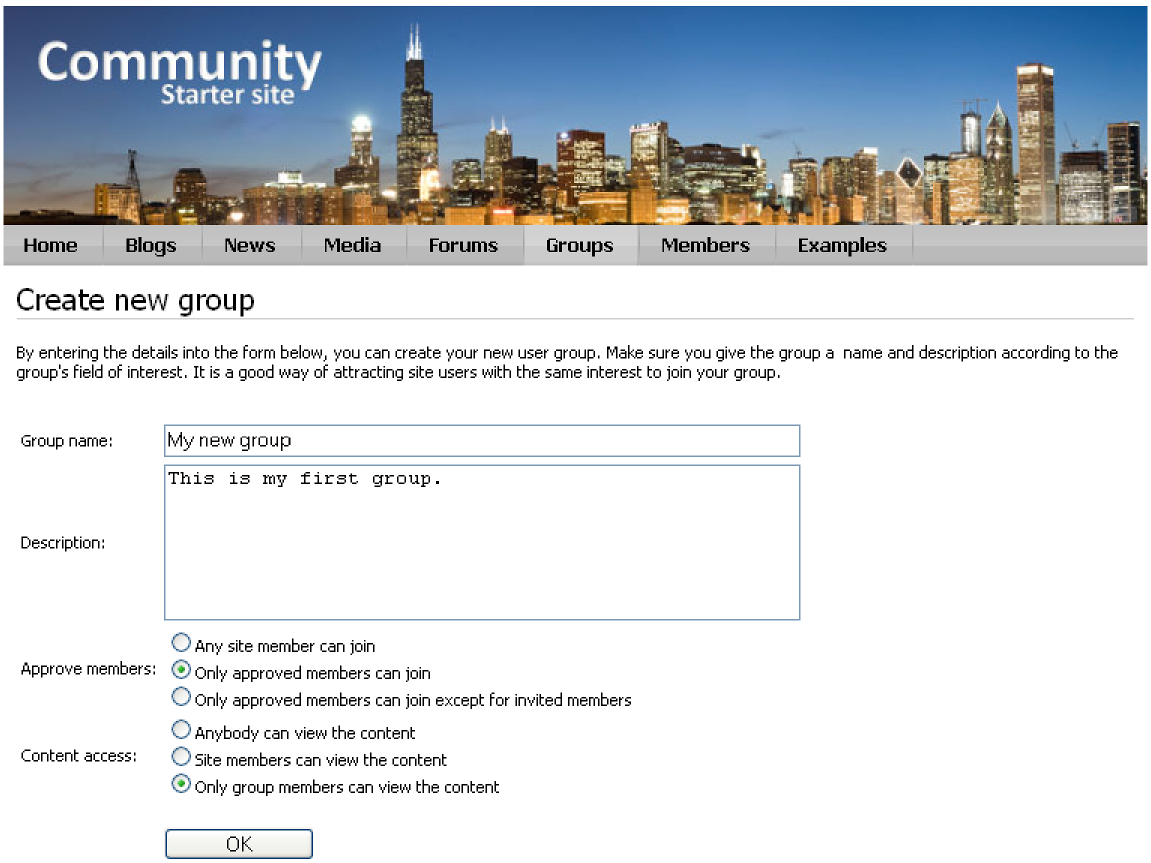


## 43.5 How site users create a new group

When a user want to create a new group on the live site using the **Group registration** web part, she has to fill in the following details:

- **Group name** - name of the group displayed in 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

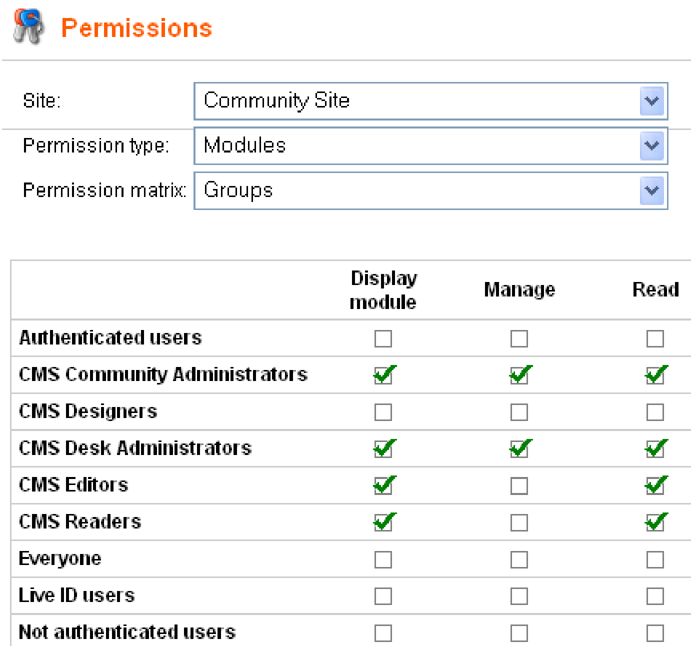
After clicking **OK**, the group will be created and group pages added to the site. In case that site administrator's approval is needed, these actions will be performed after the approval.



# 43.6 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:

- **Display module** - makes the module visible in the administration interface
- **Manage** - allows managing of groups via the administration interface
- **Read** - allows seeing the values in group settings, but does not allow to make any changes to them




| | Display module | Manage | Read |
|---|---|---|---|
| Authenticated users | ☐ | ☐ | ☐ |
| CMS Community Administrators | ☑ | ☑ | ☑ |
| CMS Designers | ☐ | ☐ | ☐ |
| CMS Desk Administrators | ☑ | ☑ | ☑ |
| CMS Editors | ☑ | ☐ | ☑ |
| CMS Readers | ☑ | ☐ | ☑ |
| Everyone | ☐ | ☐ | ☐ |
| Live ID users | ☐ | ☐ | ☐ |
| Not authenticated users | ☐ | ☐ | ☐ |

## Group pages permissions

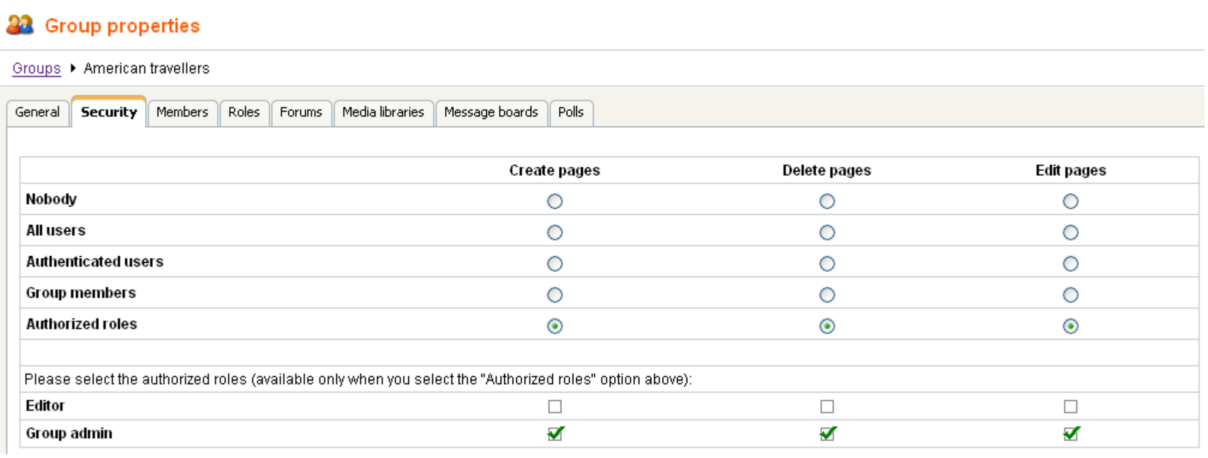
- **Create pages** - users can create group pages
- **Delete pages** - users can delete group pages
- **Edit pages** - users can edit group pages

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.

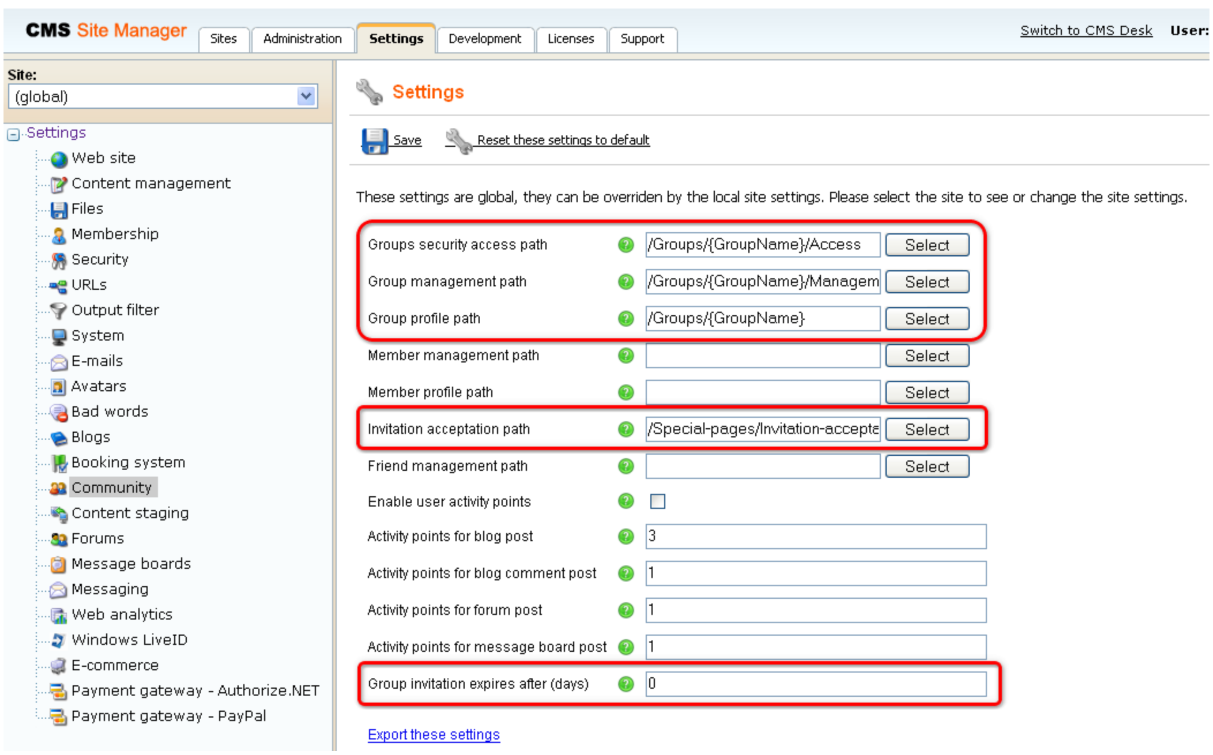


## 43.7 Settings

Setting of the groups module are located in **Site Manager -> Settings -> Community**. The following settings can be done:

- **Groups security access path** - alias path of a document to that users will be redirected when they try to access pages of a group to that 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 click the joining link in group invitation e-mail; e.g. */Special-pages/Invitation-acceptation*
- **Group invitation expires after (days)** - when some user receives 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 then, the link will be no more functional; when 0 is entered, the link will be functional permanently



# Part XX

# 44 Module Image gallery

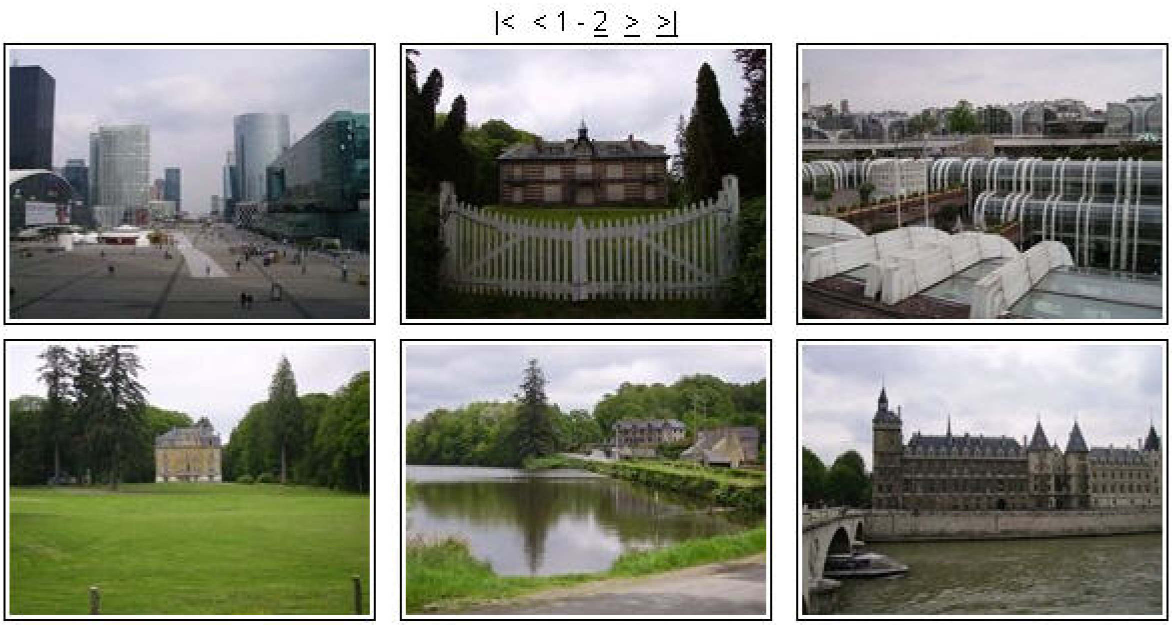
## 44.1 Overview

The module is used for effortless creating of image gallery pages. It encompasses three page templates and three web parts suitable for creating image galleries.

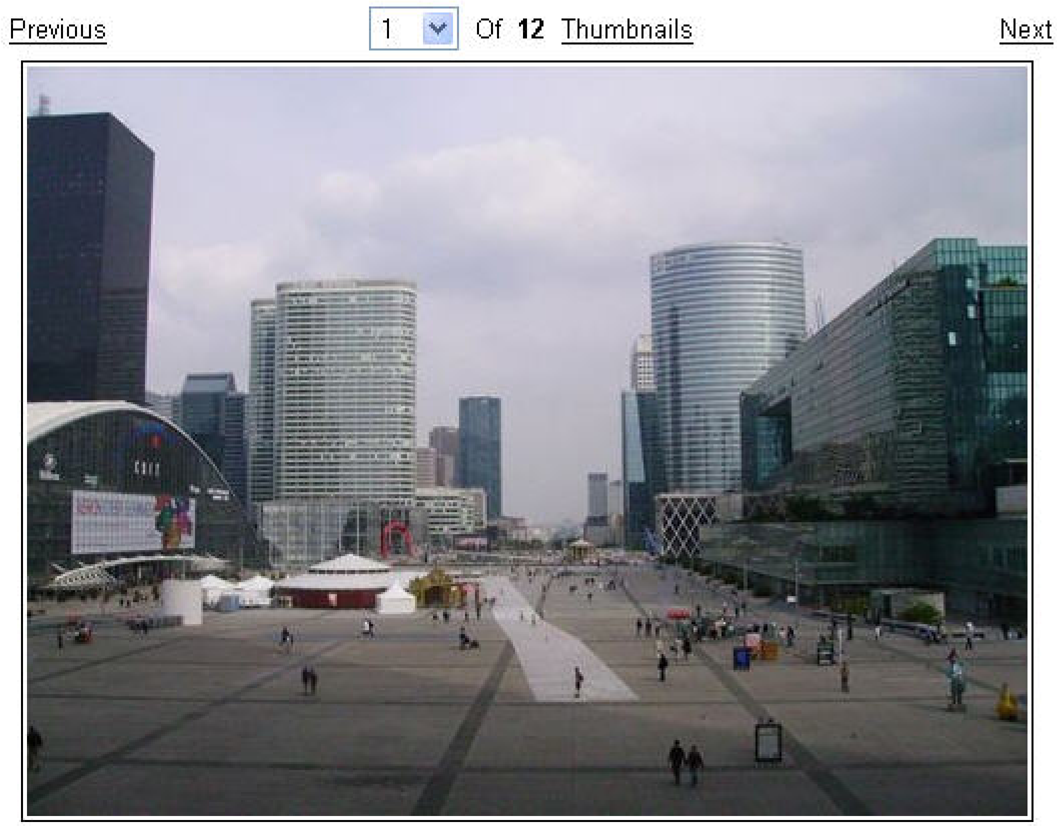
## 44.2 Available web parts

### 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:

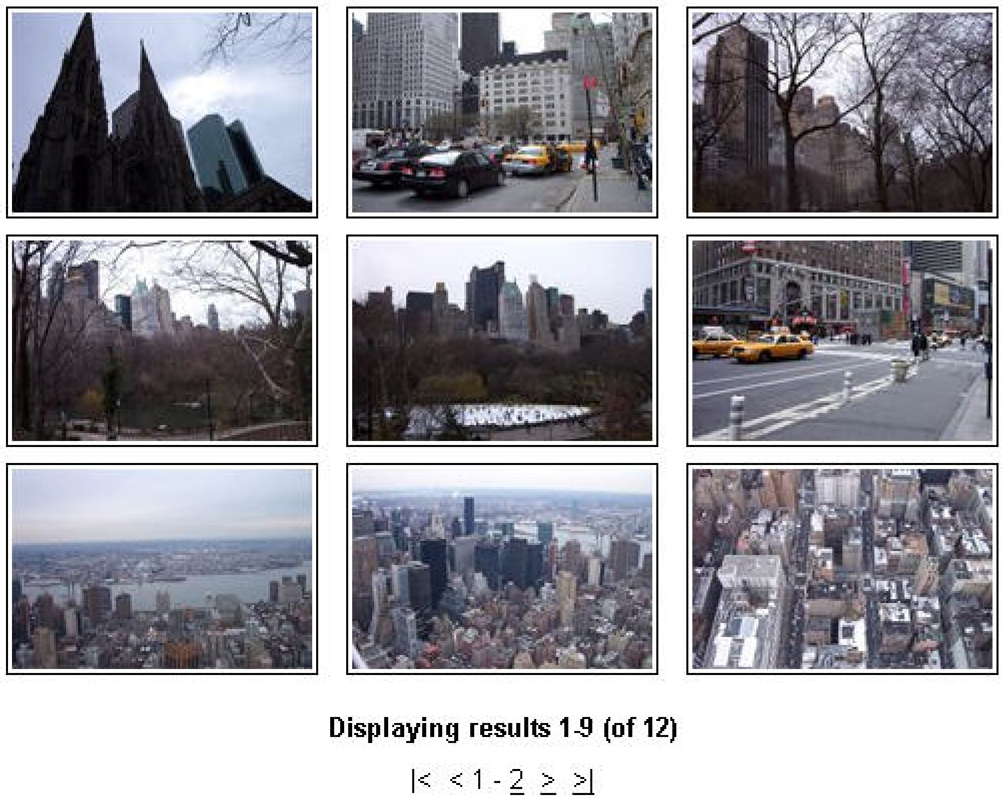



Besides the usual parameters common for all web parts, these parameters can be set to customize the appearance of the gallery:

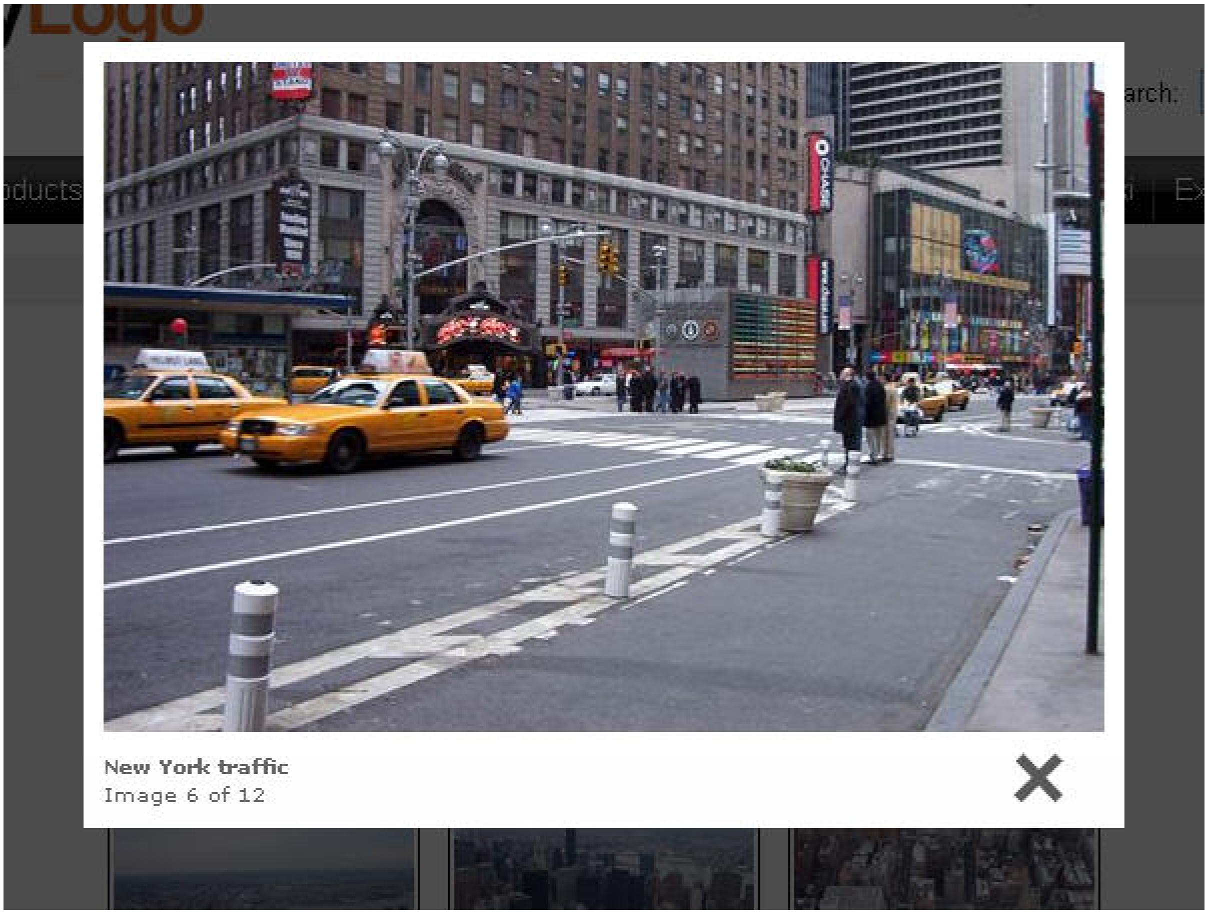
| **Transformations** | |
|---|---|
| Detail transformation | Transformation used for displaying a selected image. |
| Thumbnail transformation | Transformation used for displaying gallery thumbnails. |
| **Layout** | |
| Columns | Number of thumbnail columns in the thumbnail view. |
| Rows per page | Number of thumbnail rows per page in the thumbnail view. |
| **Paging** | |
| Paging mode | Paging parameter transfer type:<br>Query string - the paging parameter is transferred through URL<br>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 page 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 it, as you can see in the screenshot below:



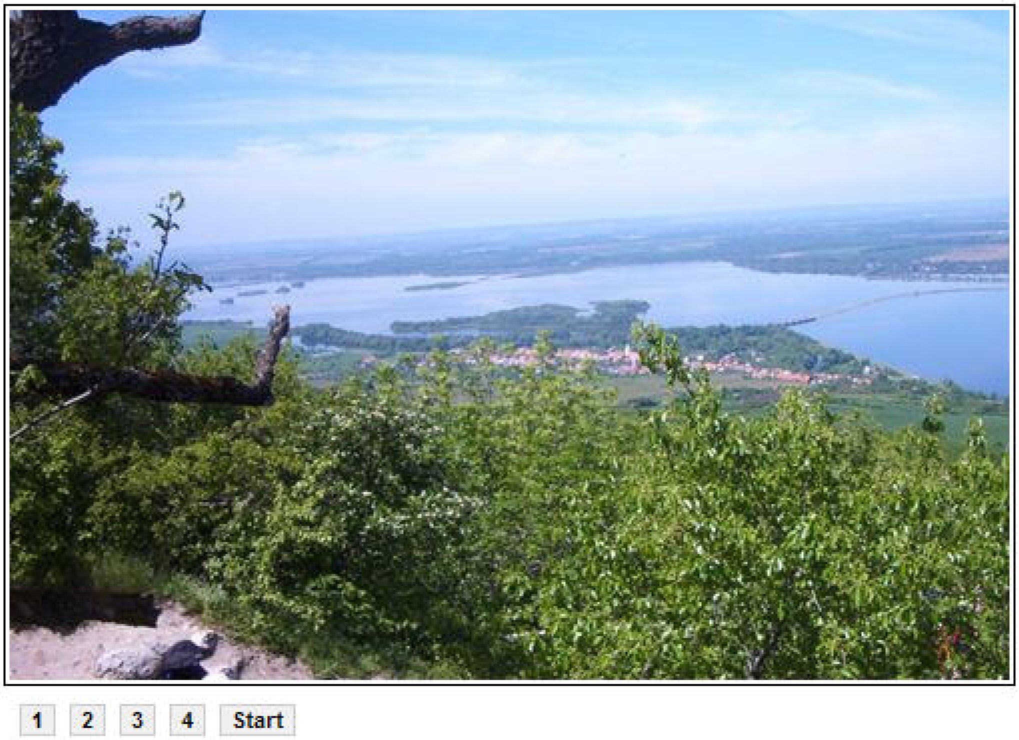

Here is a list of parameters specific for the Lightbox gallery web part:

| | |
|---|---|
| **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<br><br>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. |
| **Layout** | |
| Columns | Number of thumbnail columns in the thumbnail view. |
| Rows per page | Number of thumbnail rows per page in the thumbnail view. |

| 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:<br>Query string - the paging parameter is transferred through URL<br>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** | |
| Frame width | Width of the lightbox frame. |
| Frame height | Height of the lightbox frame. |
| Path to external scripts | URL path to the external JavaScripts required by the lightbox. |
| Overlay opacity | Opacity of lightbox background. Enter values ranging from 0 (transparent) to 1 (opaque black). |
| Animate | Enables lightbox animation. |
| 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. |

### 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 parameters of the Content slider web part:

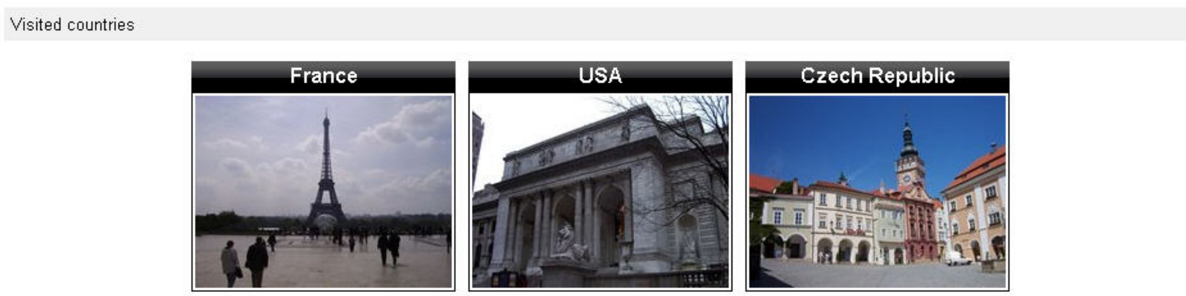
| | |
|---|---|
| **Transformations** | |
| Transformation | Transformation used for displaying the list of thumbnails. |
| Alternating transformation | Transformation used for even items in the thumbnail view. |
| Item separator | Separator displayed between thumbnails. |
| Nested repeater ID | ID of the nested CMSRepeater control as specified in transformation code. |
| Nested datalist ID | ID of the nested CMSDataList control as specified in 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 that the image will be displayed. |
| Auto start | If checked, the slide show will automatically start from the beginning. |

## 44.3 Available page templates

There are three basic page templates that can be used for image galleries.

### 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

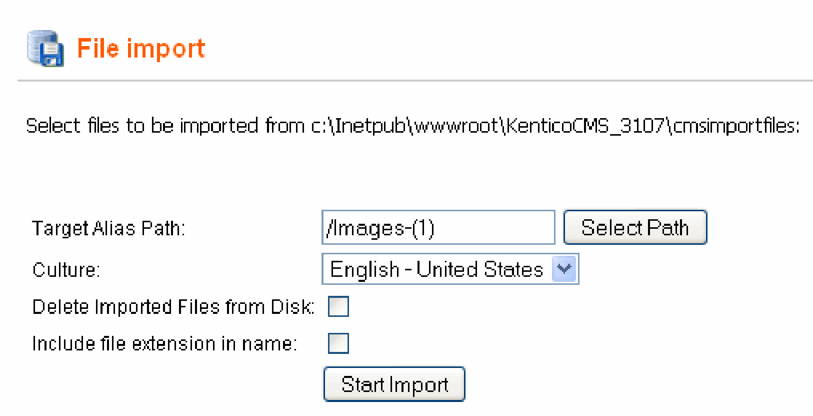
This is a basic page template used for displaying image galleries. It uses the Image gallery web part for displaying images under a given path in the content tree. See chapter Available web parts for detailed info.

### Lightbox gallery

This page template uses the Lightbox gallery web part for displaying images. See chapter Available web parts for further info.

## 44.4 Importing images

Images used in galleries are imported the same way as any other files.
Copy your files into the CMSImportFiles directory found under your website's root folder. Go to **CMSDesk -> Tools -> File import**.




On the page, select the content tree alias path that you want to save the files to. Also set the default culture of the imported files.
Checking the **Delete Imported Files from Disk** check-box causes that all files inside the CMSImportFiles will be deleted after the import is finished.
Checking the **Include file extension in name** check-box causes that file extensions will be included in the files' names in the content tree.
Finally, click the **Start import** button to import the files to the selected location.

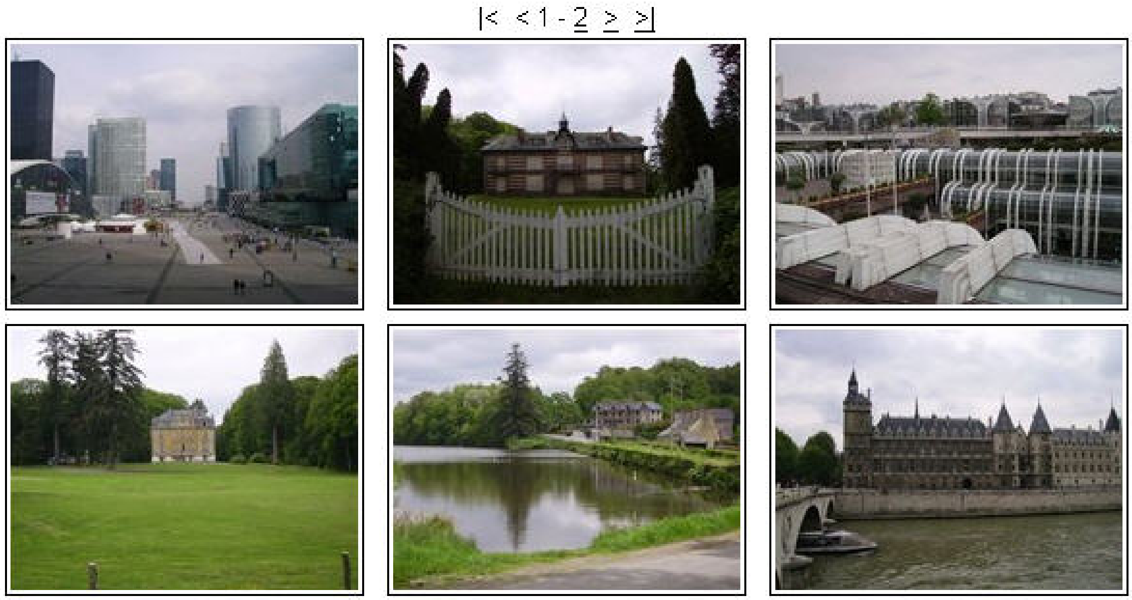
## 44.5 Transformations

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, by the means of the appropriate buttons, you can **Select** a predefined transformation from a list or **Edit** the current transformation and add a custom code to it.

Here are some examples on 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>
```

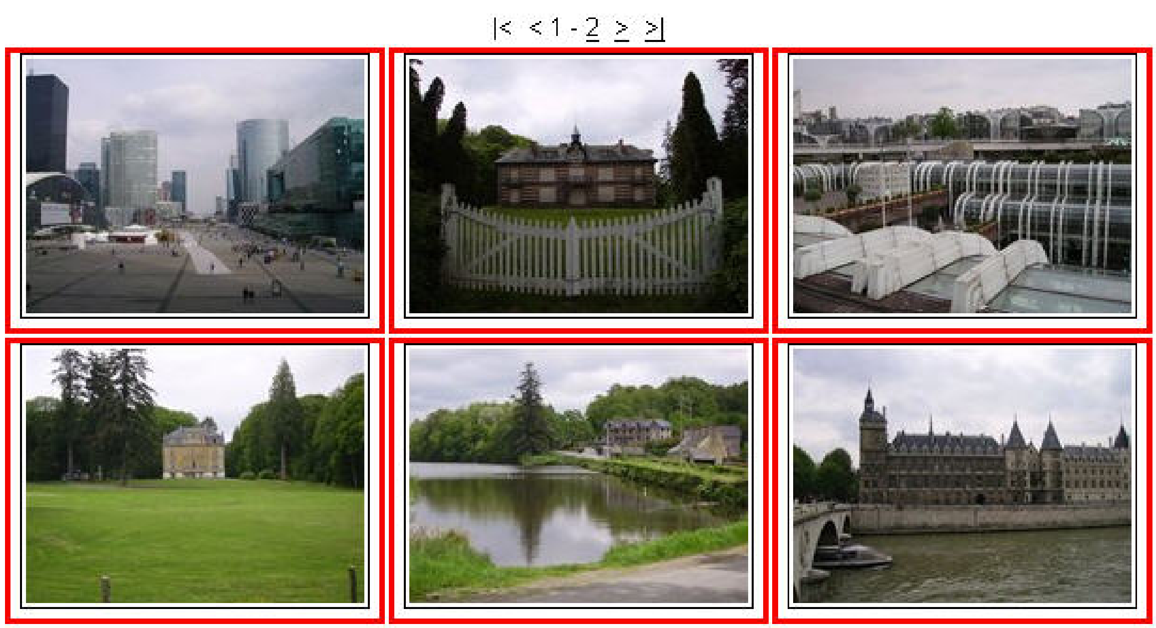
And this is how the gallery created using this transformation looks like:




In the following example, we will add red border around each thumbnail.

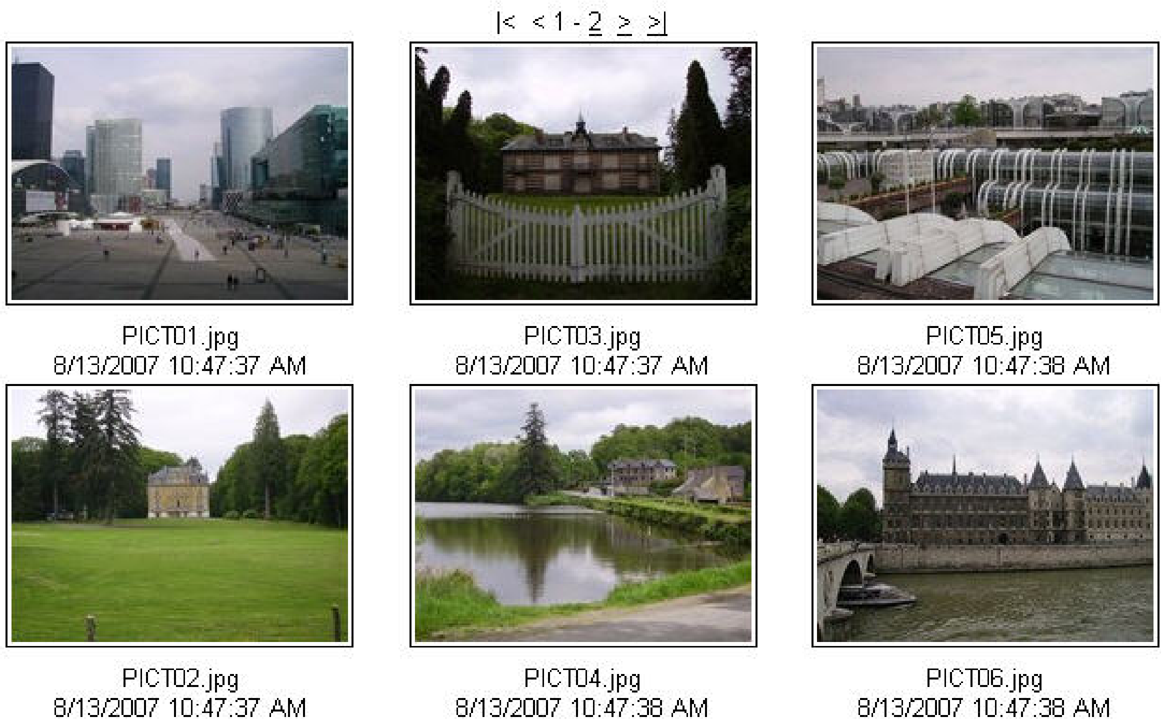
```
<div style="border: solid 3px Red">
<a href="?imagepath=<%# System.Web.HttpUtility.UrlEncode(DataBinder.Eval(Container,
"DataItem.NodeAliasPath").ToString()) %>">
<%# IfEmpty(Eval("FileAttachment"), "no image", "<img alt=\"" + Eval("FileName") + "\"
src=\"" + GetFileUrl("FileAttachment") + "?maxsidesize=180\" border=\"0\" />") %>
</a>
</div>
```

This is how the result looks like:



This example shows how to display 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 web part**

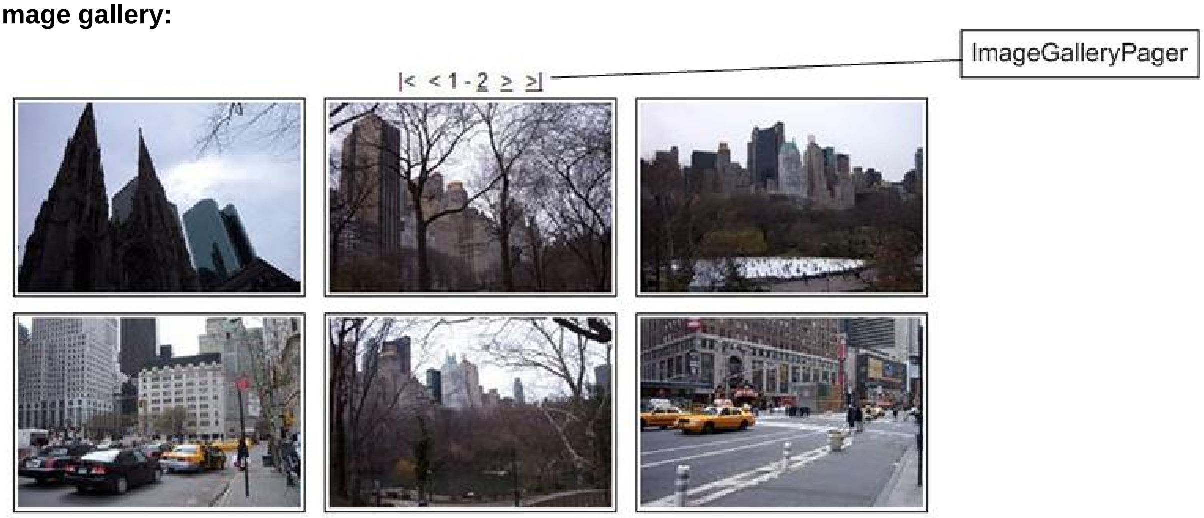
When writing a custom transformation for the Lightbox web part, it is necessary to use the 'rel' and 'rev' parameters as highlighted in the transformation code below. The 'title' parameter is used to determine the description of the image displayed in the lightbox.

```
<a href="<%# GetDocumentUrl() %>" rel="lightbox[group]" rev="<%# Eval("NodeAliasPath") %>"
title="<%# Eval("FileDescription") %>"><img src="<%# GetFileUrl("FileAttachment")
%>?maxsidesize=150"
alt="<%# Eval("FileName") %>" /></a>
```

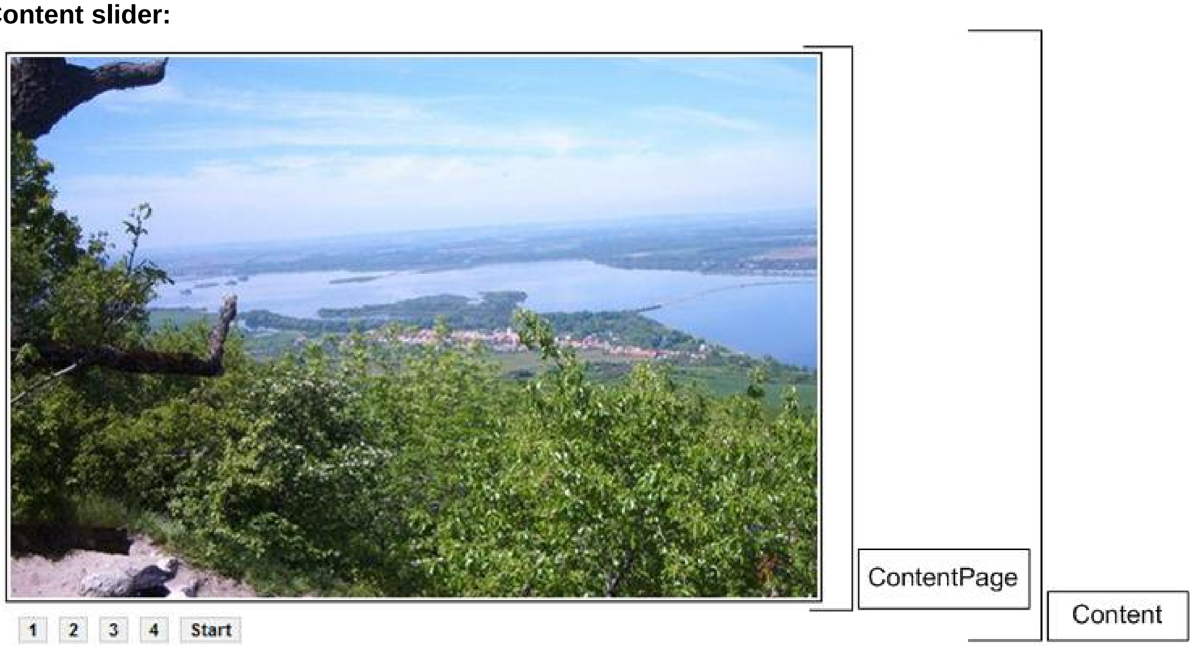
## 44.6 Design and styles

The web parts contain the following CSS classes. The style of these is controlled by the CSS styles included in your CSS stylesheet for a given site.

**Image gallery:**




**Content slider:**




The **Lightbox gallery** web part contains no CSS classes.

# Part XXX

# 45 Module Media libraries

## 45.1 Overview

The Media libraries module enables storing of different content 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 the libraries.

Media library files are stored directly in the file system, organized into folders. Physical path to the root of all libraries within the specified site is **<sitename>/media**, it is not customizable. Media library belongs either to a specified group or is global for a specified site.

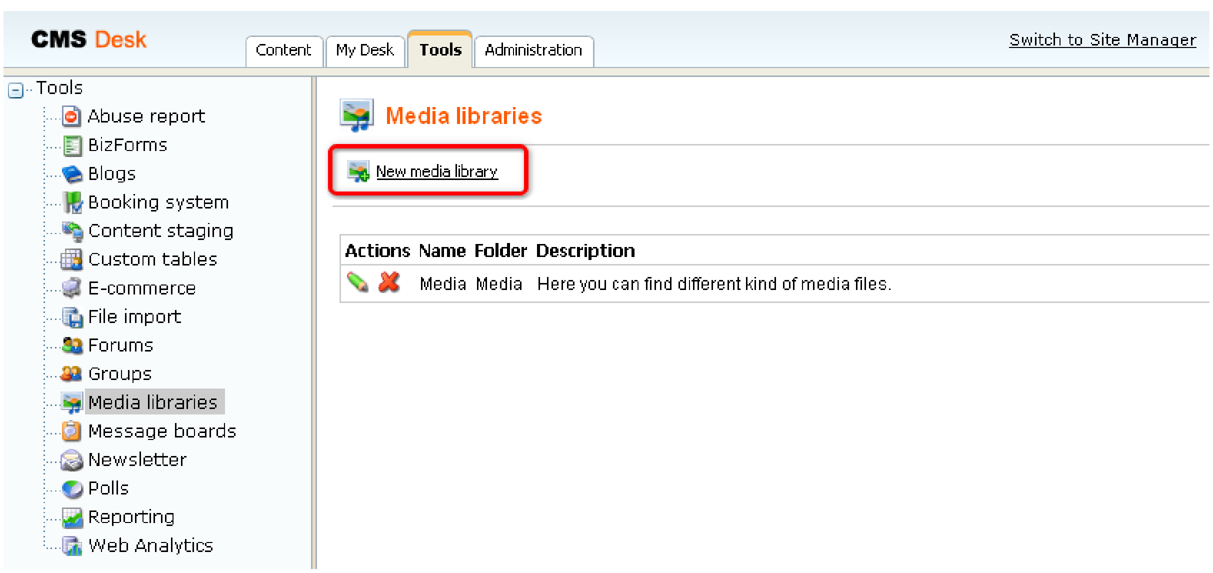
- Support for storage of large amount of files (photo galleries)
- Support for storage of large files (video files)
- Each group of users can have any number of libraries
- Files can be uploaded to the libraries externally, e.g. using FTP

## 45.2 Creating media libraries

There are two ways how media libraries can be created:

- **Live site**: Group administrators can create groups' 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 groups' media libraries in **CMS Desk -> Tools -> Groups -> Edit a group () -> Media libraries**

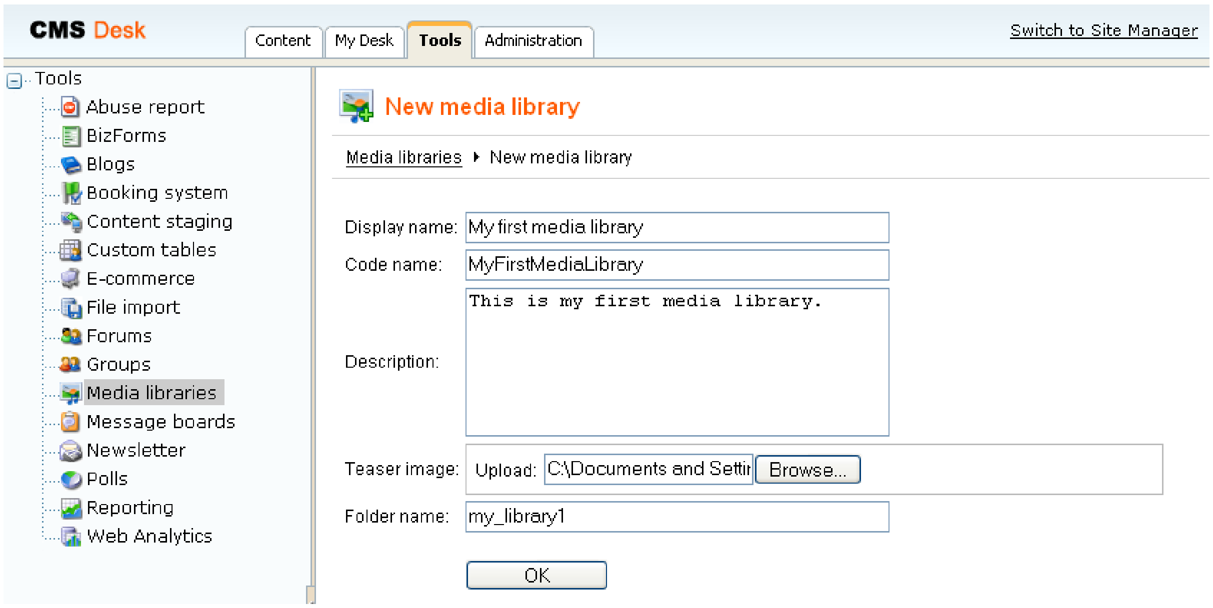
In all of these cases, media libraries can be created by clicking the **New media library** link above the list of libraries, as you can see in the screenshot below.



After clicking the link, the following details need to be entered:

- **Display name** - name of the group displayed in the administration interface and on the live site
- **Code name** - name of the group used in site code
- **Description** - text describing the media library
- **Teaser image** - image used as the media library's teaser
- **Folder name** - name of the folder where files will be stored; this folder will be created under **<web project>/<site name>\media**

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.




## 45.3 Uploading files into media libraries

There are two ways how files can be uploaded to a media library:

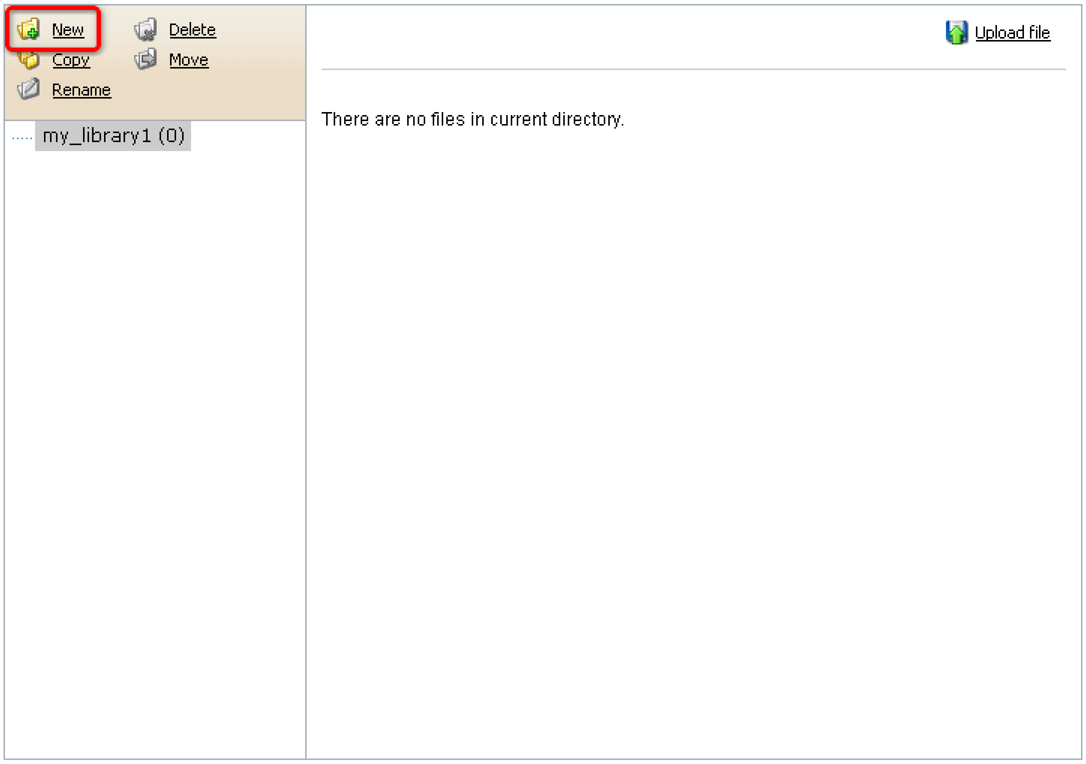
- **Via the administration interface**
- **On the live site in the Media gallery web part**
- **Externally, directly into the file system - e.g. via FTP**

In the following example, you will learn how to add files into media libraries both via the adminstration interface, on the live site and via FTP.

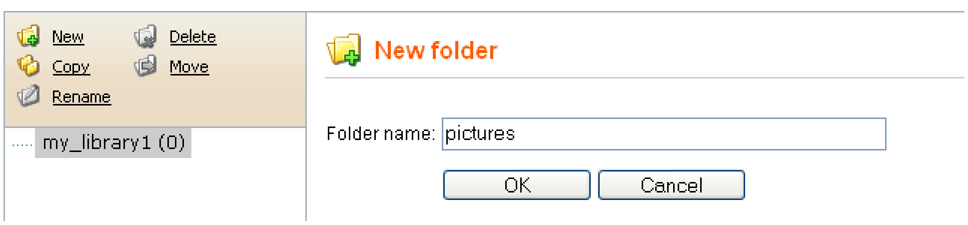
### Uploading files via the administration interface

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.

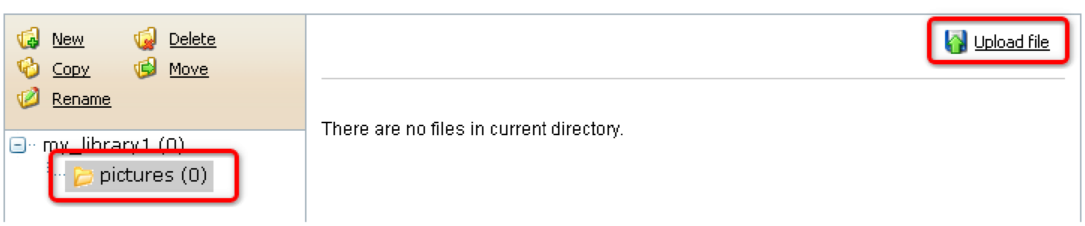
1) It is a good idea to keep your files organized in folders within the media library. To create a new folder, click the **New** link at the top left corner of the page.




2) Enter the name of the folder, e.g. *pictures*, and click **OK**.



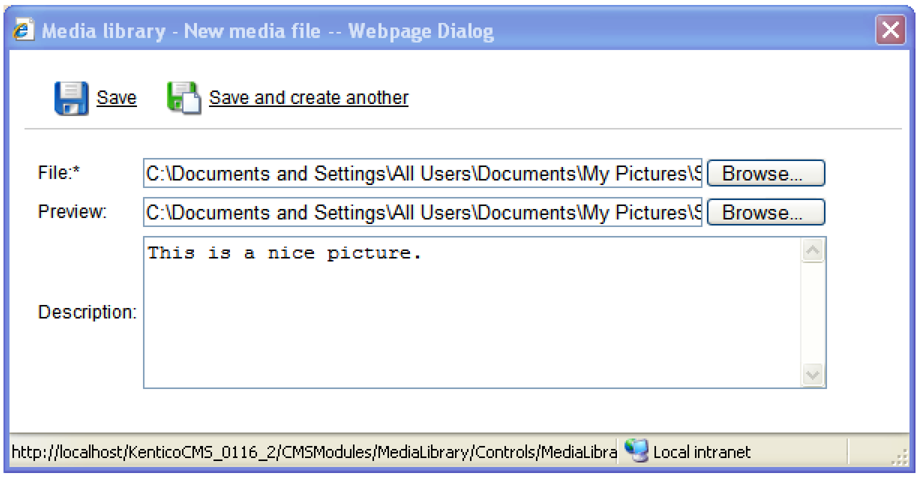

3) The folder is now created. You can upload files into the folder by clicking the **Upload file** link at the top right corner of the page.



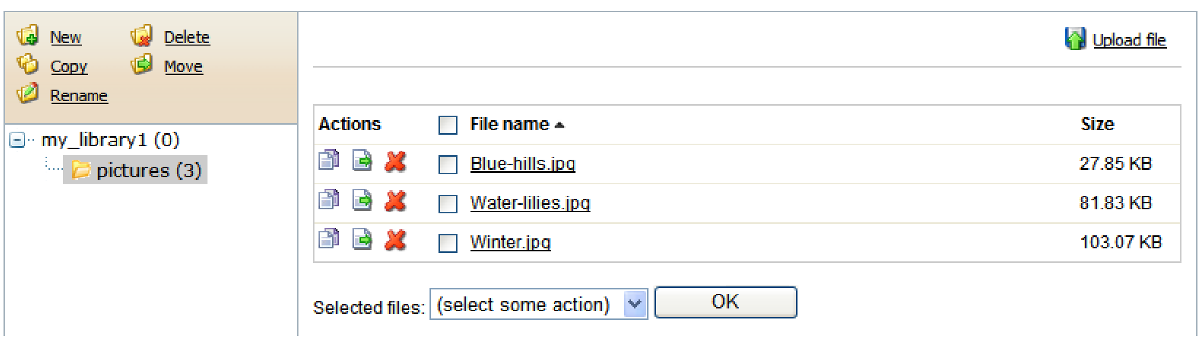
4) The following dialog appears, letting you upload the file. Enter the following details:

- **File** - path to the file you wish to upload
- **Preview** - path to the image that will be used as the file's preview; only images can be used as previews of any type of files; this field is optional
- **Description** - text describing the file

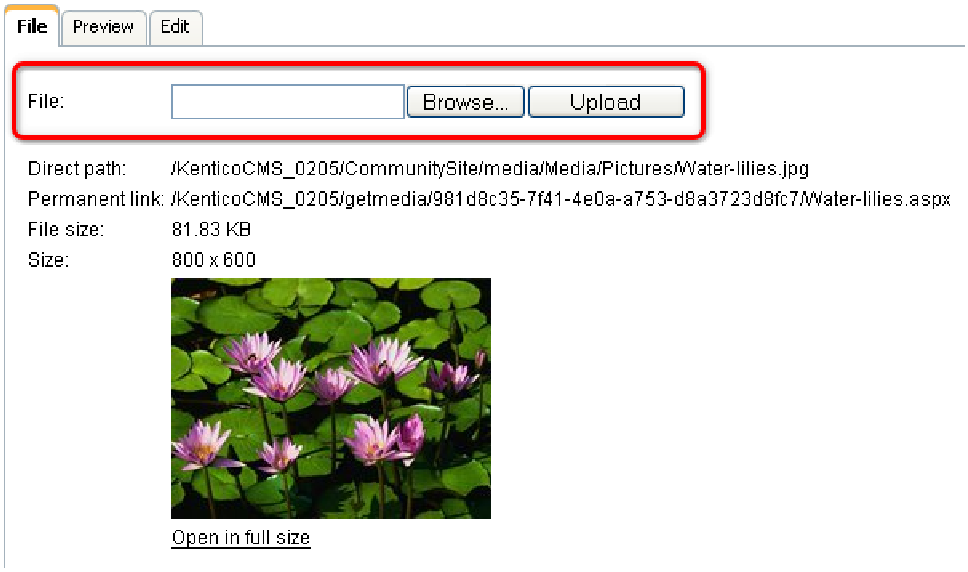
And click **Save**. If you want to upload another file right after this one, click **Save and create another**. Try uploading a few files.



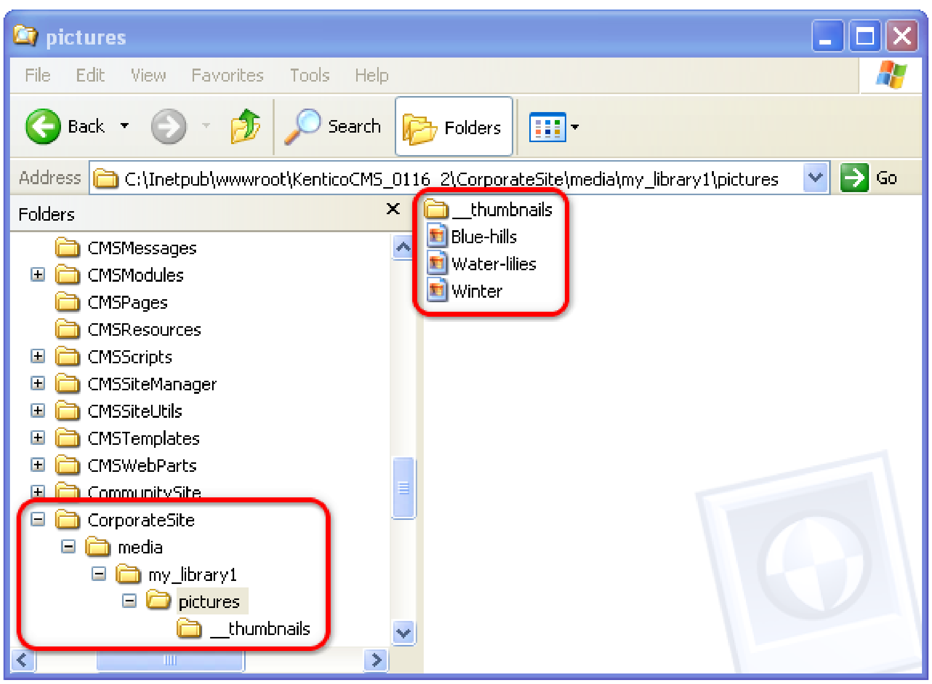

5) The files are now stored in the media library and should be displayed in the list, as you can see in the screenshot below.



6) If you click some of the uploaded files, you can **re-upload** the file directly by the controls highlighted in the screenshot below. Like this, the original file will be deleted and replaced by the newly uploaded one.




7) 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 name and location of the folder are always the same and can not be changed.

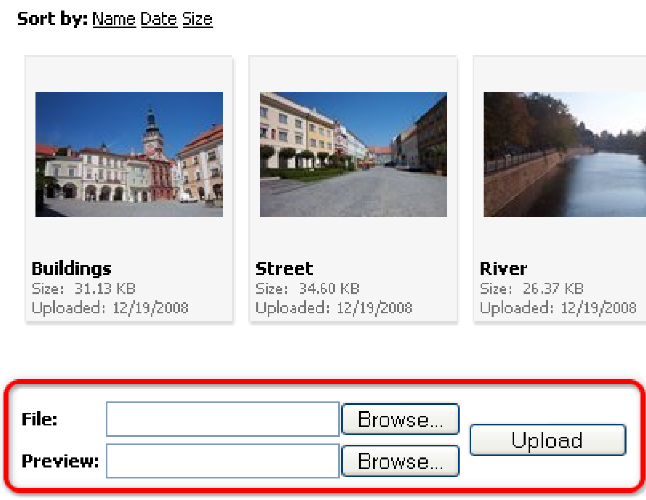


## On-site upload via the Media gallery web part

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 preview** - enables on-site upload of file previews

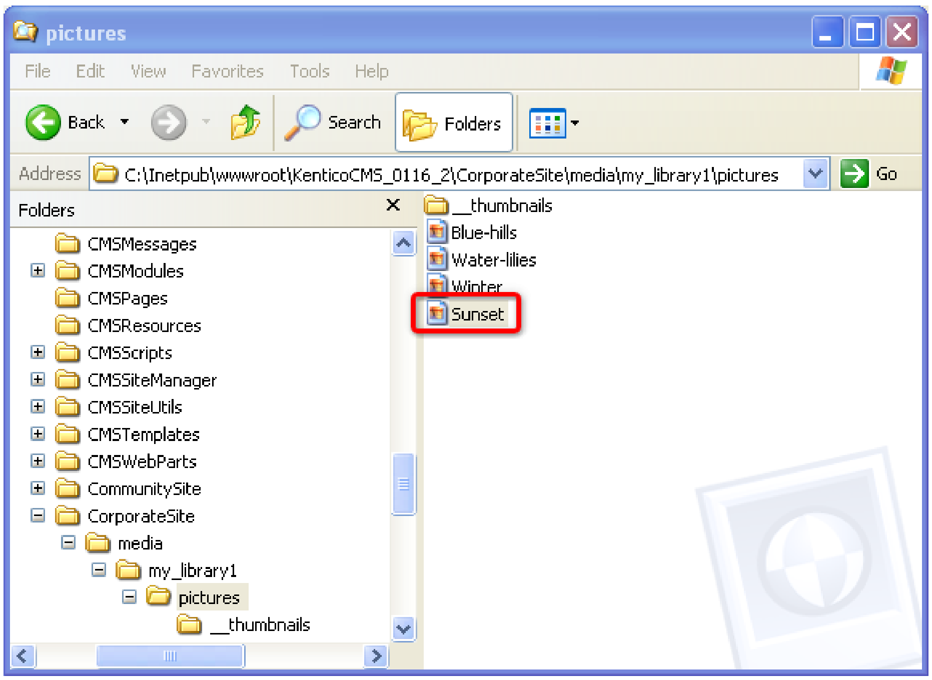
If these properties are enabled, the controls highlighted in the screenshot below will be displayed in the web part, letting users upload the files or previews:



## External upload

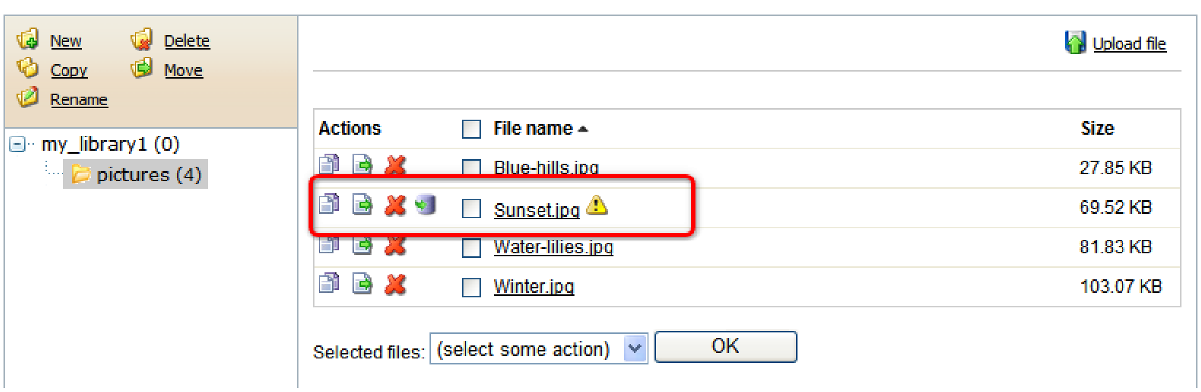
7) Files can also be uploaded **externally**, without any use of Kentico CMS administration interface or on-site web parts, e.g. using **FTP**. To do this, you simply need to copy the files into the appropriate folder of the media library.

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.




8) 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. 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.

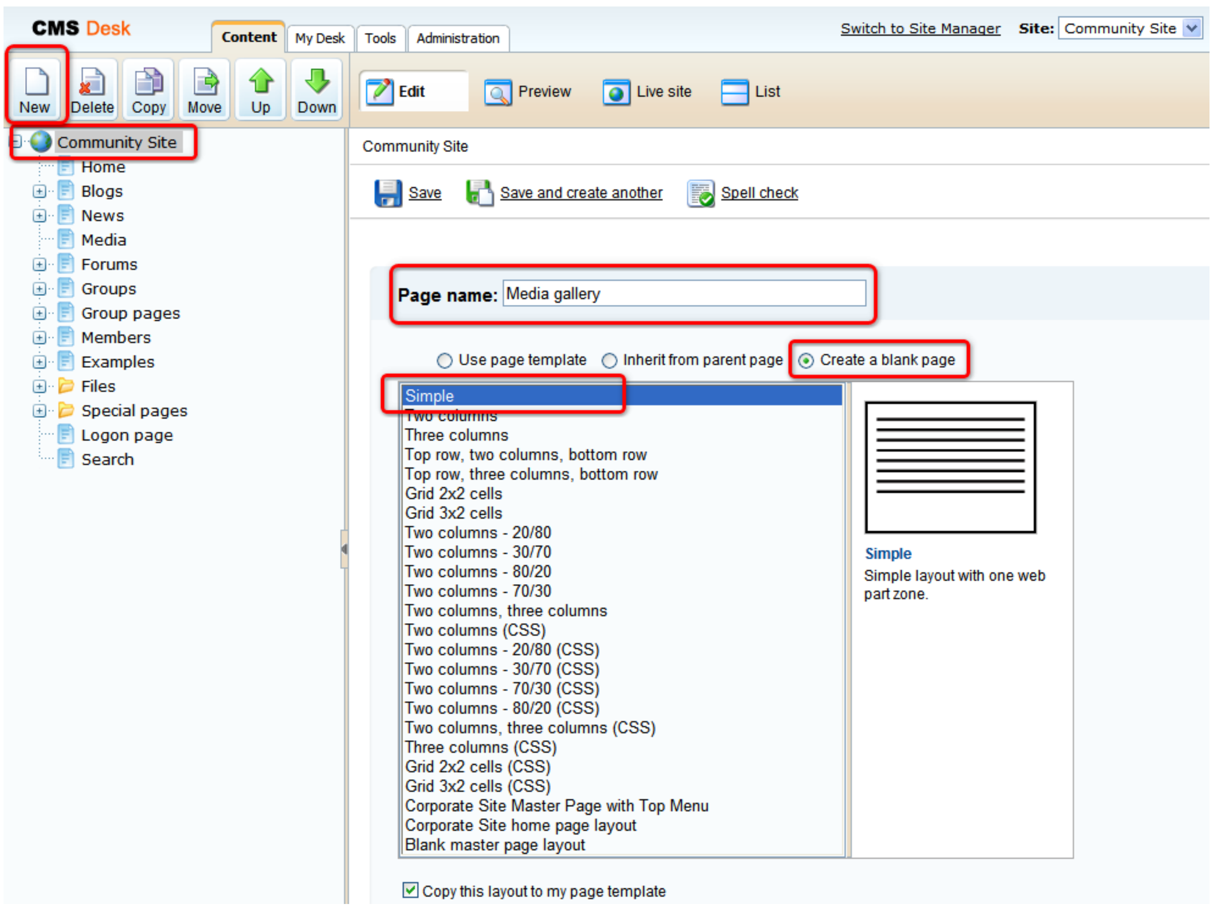
To register the file, you have to click the **Import** ( ) icon, which creates the file's record in the database. Batch file import can be done by selecting more files by the check-boxes, selecting Import from the **Selected files** drop-down list and clicking **OK**.



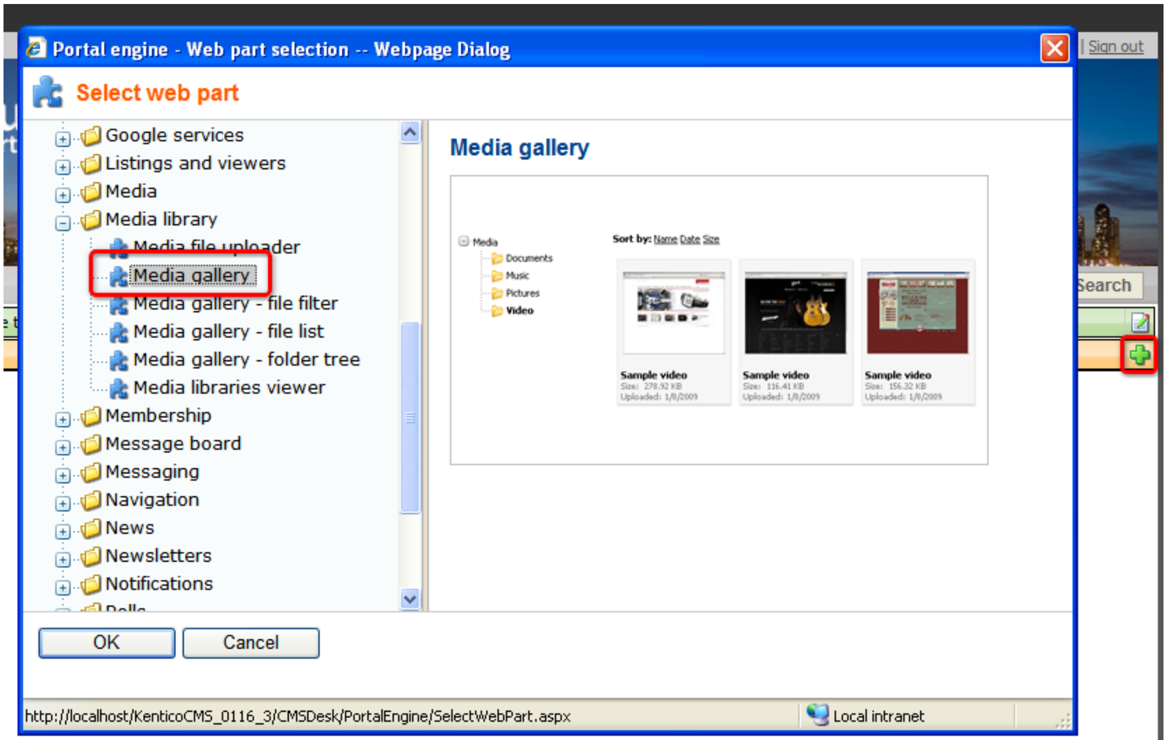
## 45.4 Displaying Media gallery content on your site

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, 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**.

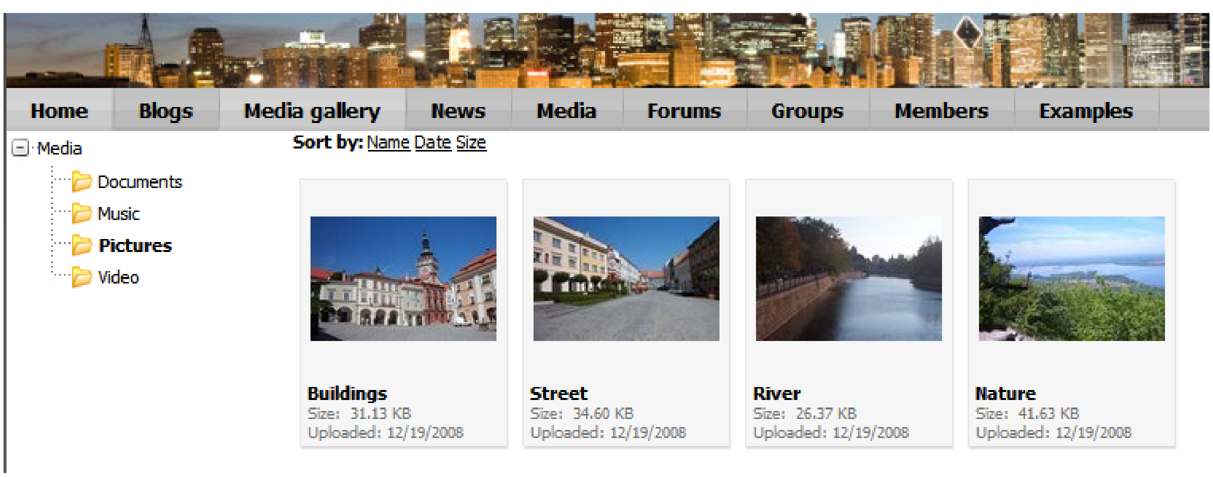



3) In the web part properties window, you only have to choose the media library that you want to display:

- **Media library** - Media

Click **OK**.

4) And that's it! If you switch to the live site, you should see the web part displaying content of the selected media library.

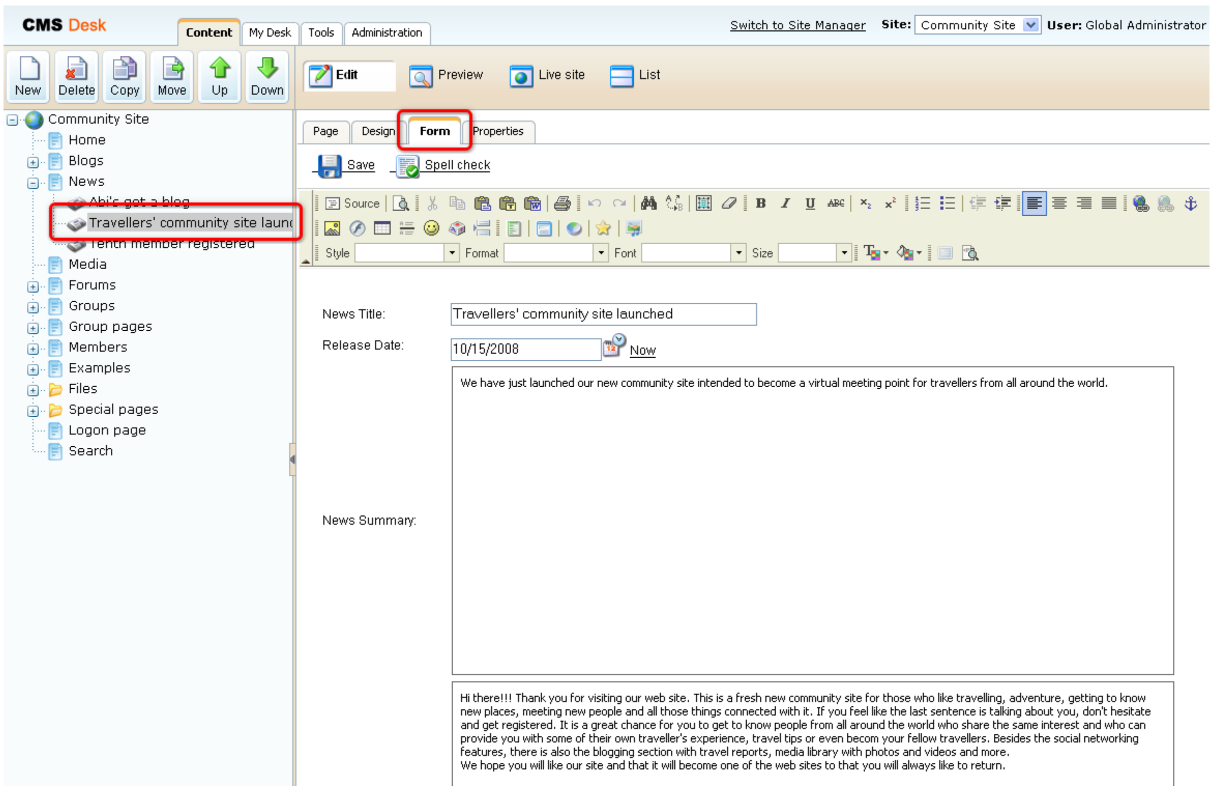


## 45.5 Support in WYSIWYG editor

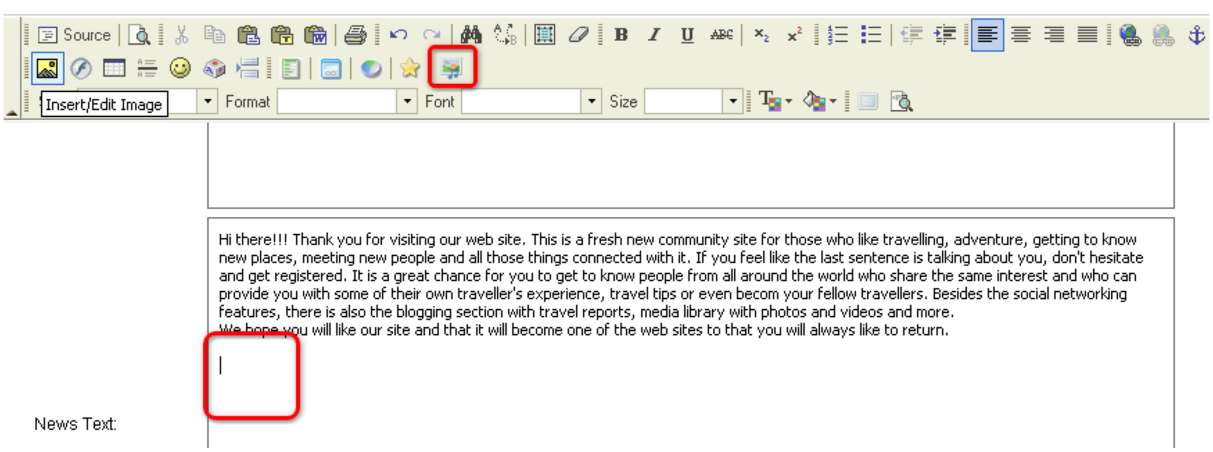
Files stored in media libraries can be inserted to documents using the built-in WYSIWYG editor.

In the following example, you will learn how to **add a video** stored in the sample media library to the **Travellers' community site launched** news item on the **Community Starter site**.

1) Go to **CMS Desk** and choose **News -> Travellers' community site launched** from the content tree. Switch to the **Form** tab.



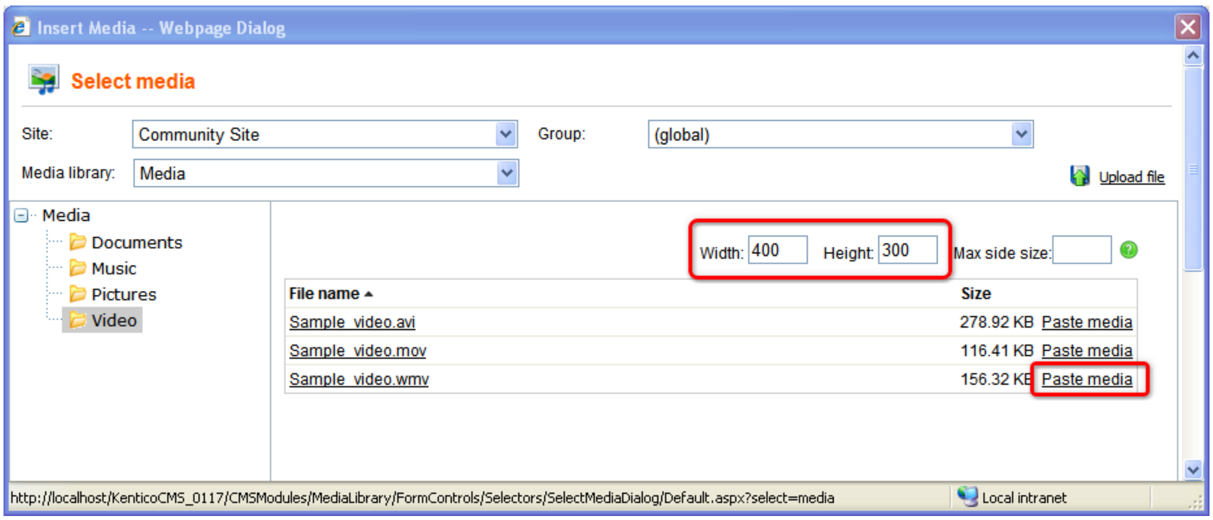
2) Place the cursor below the text in the **News Text** section, which is the place where we want to add the video, and click the **Insert media** icon.



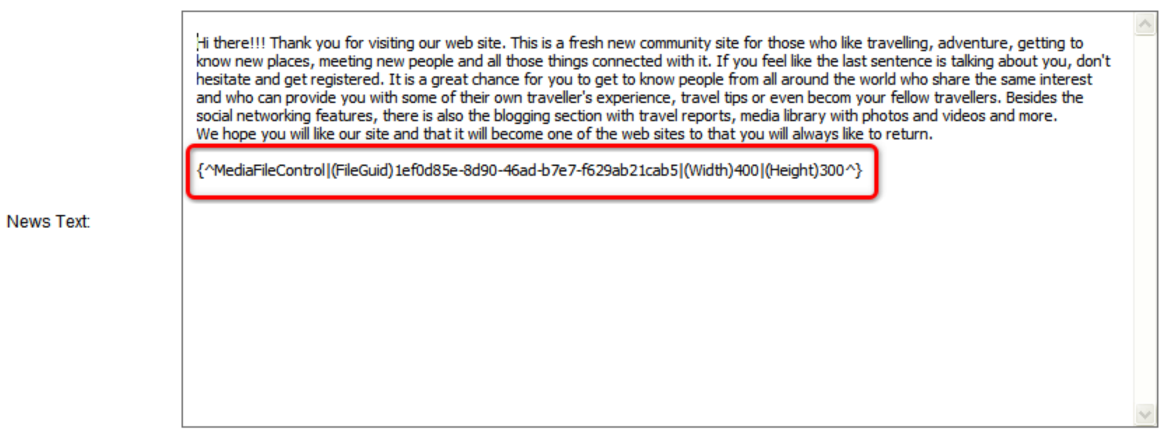
3) The **Select media** dialog pops-up. Using this dialog, you can add files from any media gallery on the site to your content. Choose the following media library:

- **Site**: Community Site
- **Group**: Global
- **Media library**: Media

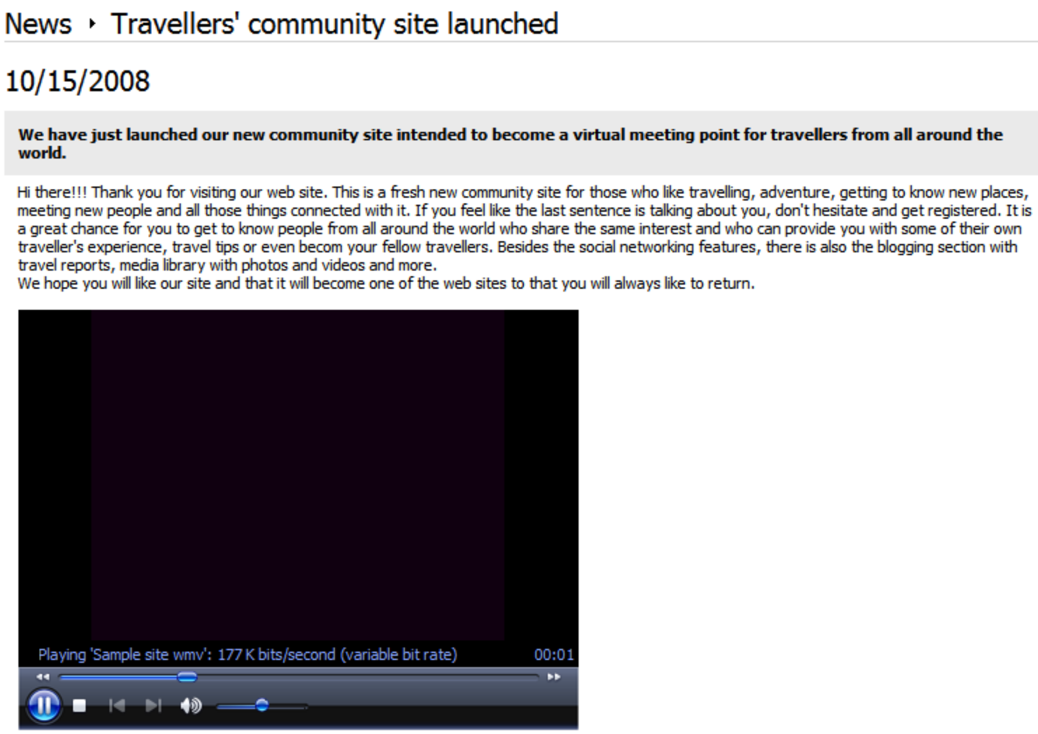
From the folder structure, choose **Video**. Enter **Height: 400** and **Width: 300** and click the **Paste media** link next to the **Sample_video.wmv** file.




The dialog closes and a macro is added to the placement of the cursor below the text.



4) If you switch to the live site now, you should see a player with the added video below the news' text.




## 45.6 Available web parts

Below, you can find a list of web parts that come with the Media libraries module and a brief description of each of them. You can find a detailed description of their properties in **Kentico CMS Web Parts** reference guide.

### Web parts -> Media library

- **Media file uploder**: enables users to upload files to the specified media library
- **Media gallery**: displays content of the media library (folder tree, file list and sorting at once)

<u>**Gallery separate web parts:**</u>

- **Media gallery – folder tree**: Displays media library folder tree structure and enables user to browse it
- **Media gallery – file list**: Displays media files from the specified folder (list + detail)
- **Media gallery – file filter**: Displays simple sorting control which enables users to sort files

\+ **Media files data source**

- **Media libraries viewer**: Displays list of media libraries

\+ **Media libraries data source**

### Web parts -> Data sources

- **Media files data source**: Creates data source from media files.

- **Media libraries data source**: Creates data source from media libraries.

**Webparts -> Community**

- **Group media libraries**: Displays administration interface for managing media libraries of the specified group
- **Group media libraries viewer**: Displays list of media libraries of the specified group

+ **Group media libraries data source**

## 45.7 Supported file types

The following file types are supported and can be played or displayed on your site:

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

**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.

## 45.8 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 this, you also need to add the following highlighted code to the end of your 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>
```

# 45.9 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

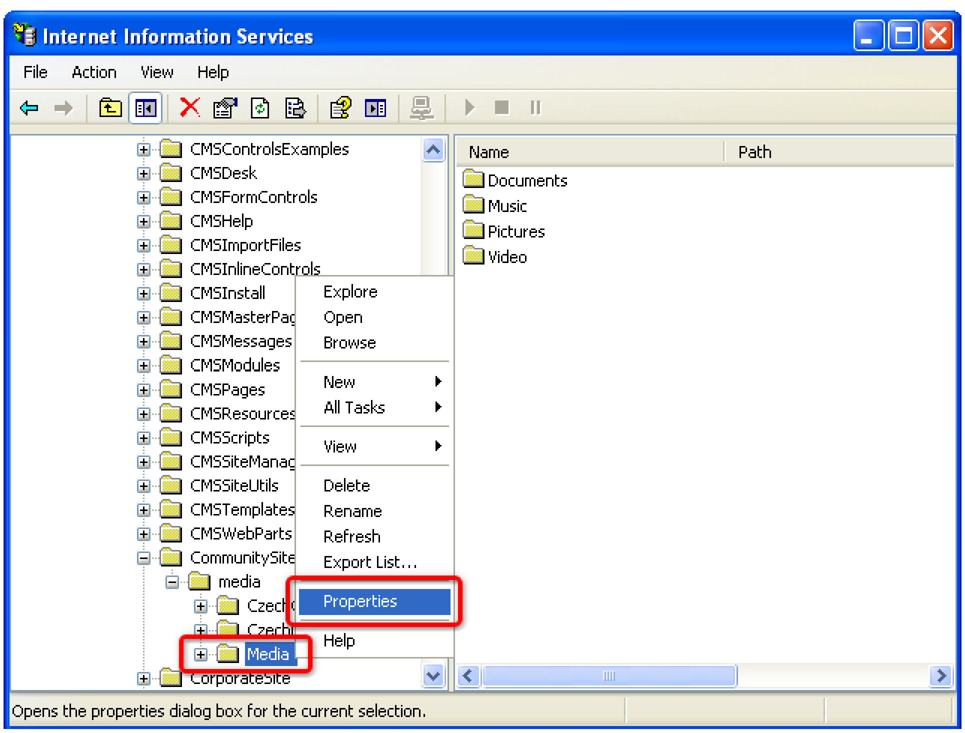
Secured media libraries allow viewing of their content only to members of authorized roles or only to authenticated users, based on the settings made on the library's Security tab. Secured libraries are also slower than the non-secured ones, as permission checking involves some processing overhead.

To set up a media library to behave as a secured library, you have to take the following steps:

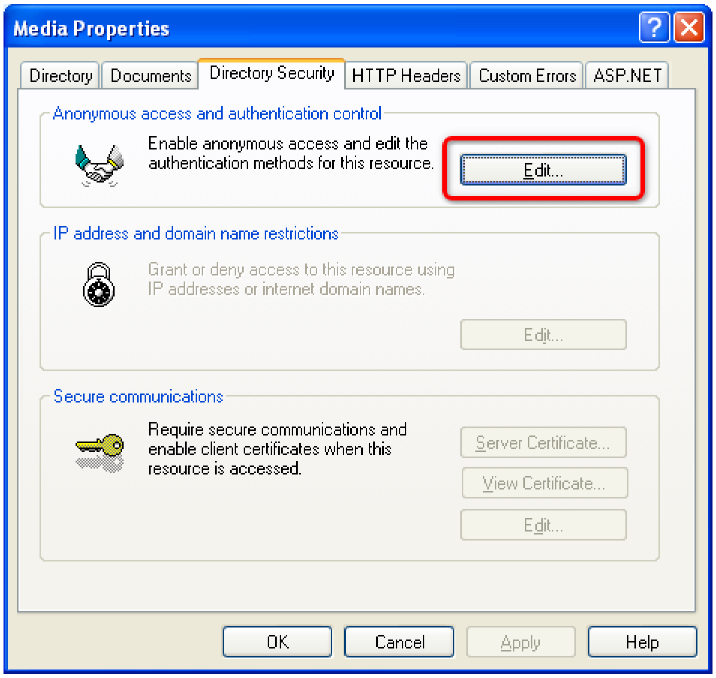
### 1) IIS setup

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.

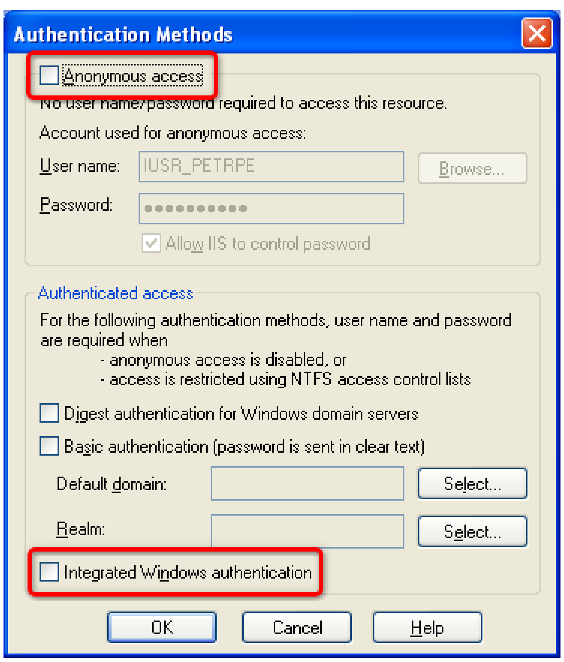
a) Locate the media library folder in IIS tree, right-click it and choose **Properties**.



b) Switch to the **Directory security** tab and click **Edit**.

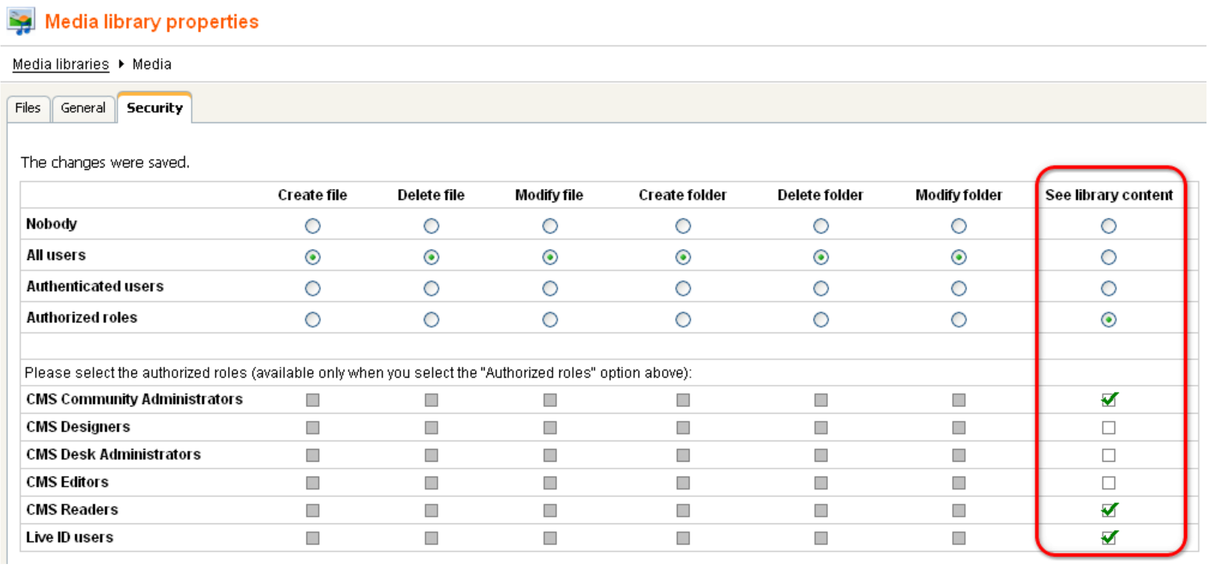



c) Uncheck the **Anonymous access** and **Integrated windows authentication** check-boxes and click **OK**.



### 2) Media library security settings

You have to assign the **See media 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 the transformations, on the sample **Community Starter site**, in the **Media** section.

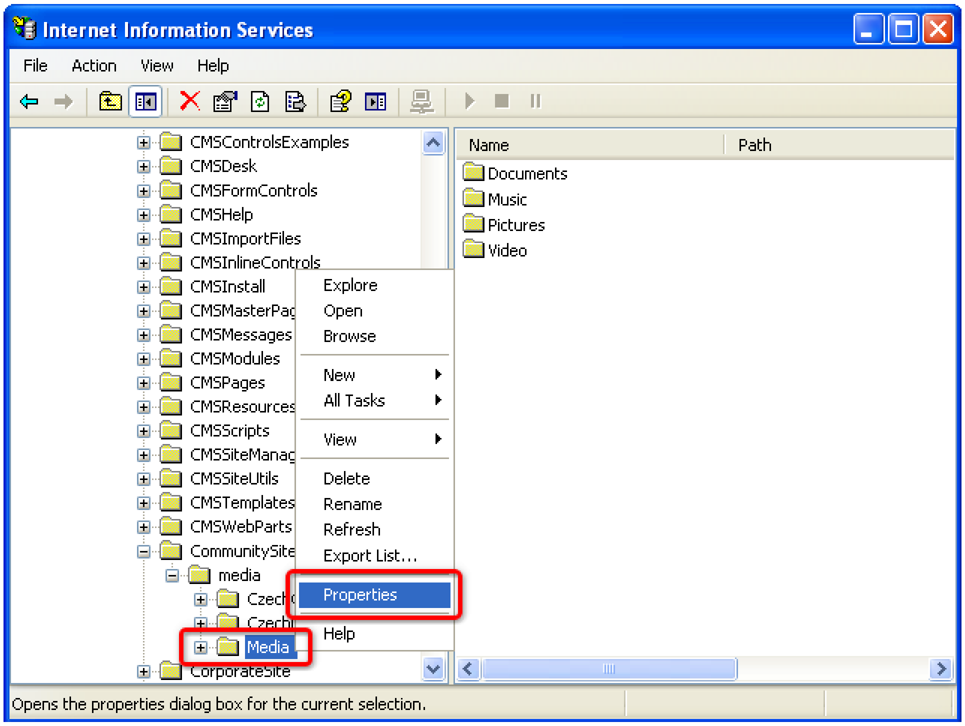
## Non-secured libraries

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.

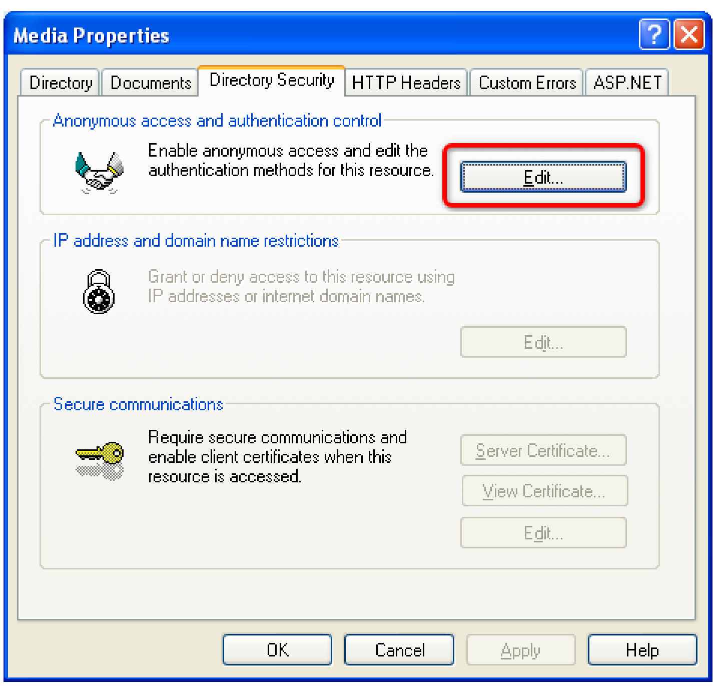
## 1) IIS setup

You have to **set up your IIS** so that files in the library can be accessed directly by anonymous users.

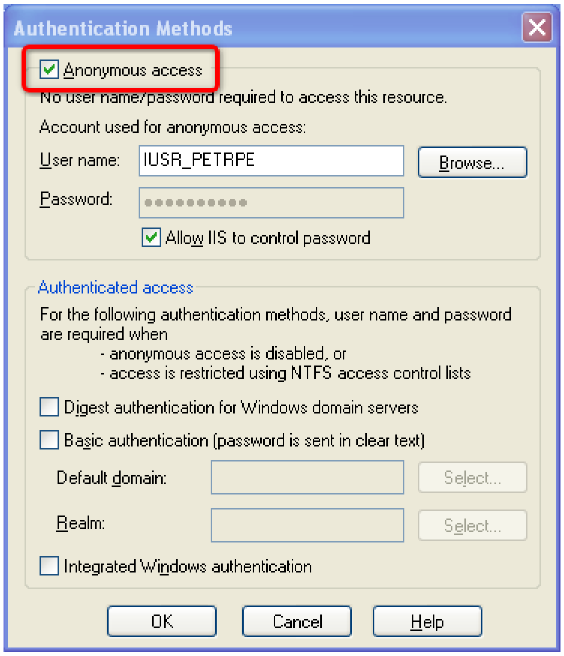
a) Locate the media library folder in IIS tree, right-click it and choose **Properties**.




b) Switch to the **Directory security** tab and click **Edit**.

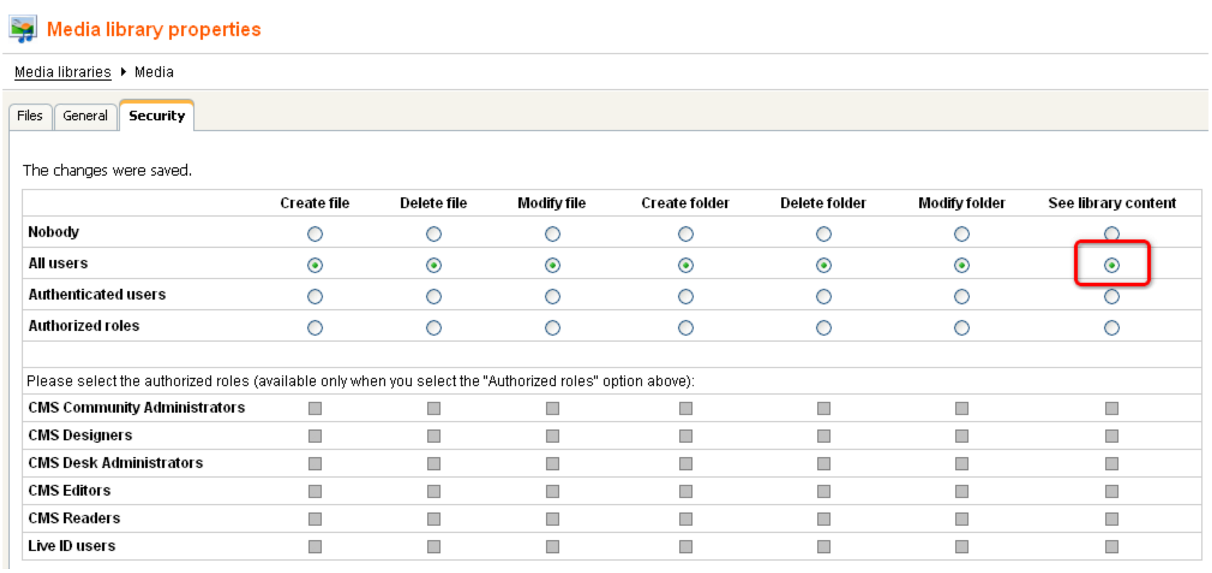


c) Make sure that the **Anonymous access** check-box is enabled and click **OK**.

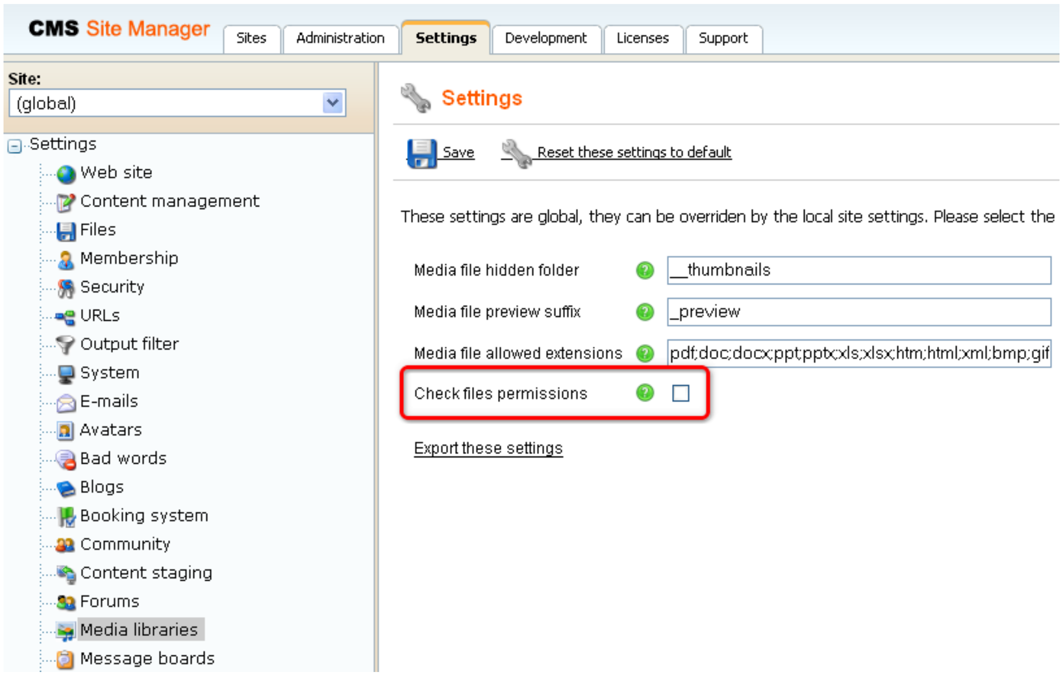



## 2) Media library security settings

You have to assign the **See media 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 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** should can be obtained using direct links. **Download links** can be obtained directly too.

## 45.10 Names of files and previews

When a file is being uploaded to a media library, users can upload the file itself and its preview. The **preview** will be displayed in the files list in the **Media gallery** web part. Previews are typically used if you want to upload some non-image file and attach an image which will be displayed as the file's preview in the file list. For image files, the preview can, bud doesn't have to be done as the image itself will be used for this purpose.

When the preview image or the image itself is larger than the required preview image size, its **thumbnail** will be created with the required size. Both previews and thumbnails 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 library hidden folder**.

**Format of media library files on the disk**

- **Files**: <filename>.<fileextension>
- **File thumbnails**: <hidden folder>/<filename>_<fileextension>_<width>_<height>.<extension>

- **Previews**: <hidden folder>/<filename>_<fileextension><previewsuffix>.<previewextension>
- **Preview thumbnails**: <hidden folder>/<filename>_<fileextension><previewsuffix>_<width>_<height>.<previewextension>

**Example**

If you upload the following files:

- **Uploaded file**: MyImage.jpg
- **Uploaded preview**: MyPhoto.bmp

with the following settings defined in **Site Manager -> Settings -> Media libraries**:

- **Media file hidden folder**: __thumbnails
- **Media file preview suffix**: _preview

the uploaded files will be stored in the following locations in the media library folder, with the following names:

- **File**: MyImage.jpg
- **File thumbnail**: __thumbnails/MyImage_jpg_100_200.jpg

- **Preview**: __thumbnails/MyImage_jpg_preview.bmp
- **Preview thumbnail**: __thumbnails/MyImage_jpg_preview_20_30.bmp

## 45.11 Site restarts

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 -> 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 don't allow users to further modify it.

## 45.12 Using custom media player/viewer

If you want to customize the media player/viewer used by the media gallery, you need to modify the **Media file preview** control, which is used for displaying all types of files.

**Control location**: ~\CMSModules\MediaLibrary\Controls\LiveControls\MediaFilePreview.ascx.cs

The customization needs to be done in the **Page_PreRender** method, in the following block of code:

```
if (this.DisplayActiveContent)
{
        if (ImageHelper.IsImage(mfi.FileExtension) && File.Exists(mfi.FilePath))
        {
            // Images are always processed by secure page
            if (!this.UseSecureLinks)
            {
                url = MediaFileInfoProvider.GetMediaFileAbsoluteUrl(si.SiteName,
mfi.FileGUID, mfi.FileName);
            }
             this.ltlOutput.Text = MediaHelper.GetImage(url, Width, Height, MaxSideSize,
mfi.FileImageWidth, mfi.FileImageHeight, mfi.FileDescription);
        }
        else if (ImageHelper.IsFlash(mfi.FileExtension))
        {
            this.ltlOutput.Text = MediaHelper.GetFlash(url, this.Width, this.Height);
        }
        else if (MediaHelper.IsAudio(mfi.FileExtension))
        {
            this.ltlOutput.Text = MediaHelper.GetAudio(url, this.Width, this.Height,
mfi.FileExtension);
        }
        else if (MediaHelper.IsVideo(mfi.FileExtension))
        {
            this.ltlOutput.Text = MediaHelper.GetVideo(url, this.Width, this.Height,
mfi.FileExtension);
        }
        else
        {
            this.ltlOutput.Text = MediaHelper.ShowPreviewOrIcon(mfi, this.Width, this
.Height, this.MaxSideSize, this.PreviewSuffix, this.IconSet);
        }
}
else
{
    this.ltlOutput.Text = MediaHelper.ShowPreviewOrIcon(mfi, this.Width, this.Height, this
.MaxSideSize, this.PreviewSuffix, this.IconSet);
}
```

In this section of the code, you can define your custom output HTML for the defined file types (images, videos, audio files, flash, etc.), which ensures correct displaying of the defined file types (displays the image, plays the video, etc.).

Below, you can find descriptions on how to perform some sample customization of the control:

### Play MP3 files using Windows Media Player

To play MP3 files using Windows Media Player, you need to change the following line:

```
this.ltlOutput.Text = MediaHelper.GetAudio(url, this.Width, this.Height,
mfi.FileExtension);
```

to:

```
this.ltlOutput.Text = MediaHelper.GetVideo(url, this.Width, this.Height,
mfi.FileExtension);
```

This is a workaround, as the GetVideo method returns Windows Media Player for all non-supported video formats.

### Display or play custom file types

The following example shows the process of enabling playing of FLV videos using the JW FLV Player (see http://www.longtailvideo.com/players/jw-flv-player/).

1) you need to buy the appropriate license (for commercial purposes, for the required domain)
2) register the **swfobject.js** JavaScript (via the **General -> Java script** web part)
3) add your custom condition - add the following code:

```
else if (mfi.FileExtension.ToLower() == ".flv")
{
   this.ltlOutput.Text = // HERE COMES HTML CODE FOR INSERTING FLV PLAYER (see example
bellow)
}
else if (MediaHelper.IsAudio(mfi.FileExtension))
```

before this line:

```
else if (MediaHelper.IsAudio(mfi.FileExtension))
```

this is the example inserted HTML code:

```
"<script type='text/javascript'>
   var s1 = new SWFObject('player.swf','player','" + Width + "','" + Height + "','9');
   s1.addParam('allowfullscreen','true');
   s1.addParam('allowscriptaccess','always');
   s1.addParam('flashvars','file=" + url + "');
   s1.write('preview');
</script>";
```

### Use Lightbox for displaying images

This modification allows viewing images using Lightbox (see http://www.lokeshdhakar.com/projects/lightbox2/ or the **Examples -> Webparts -> Listings and viewers -> Lightbox** page on the sample **Corporate site**).

1) Register the required JavaScripts, e.g. using the **General -> Java script** web part.

2) Change the following piece of code:

```
this.ltlOutput.Text = MediaHelper.GetImage(url, Width, Height, MaxSideSize,
mfi.FileImageWidth, mfi.FileImageHeight, mfi.FileDescription);
```

to:

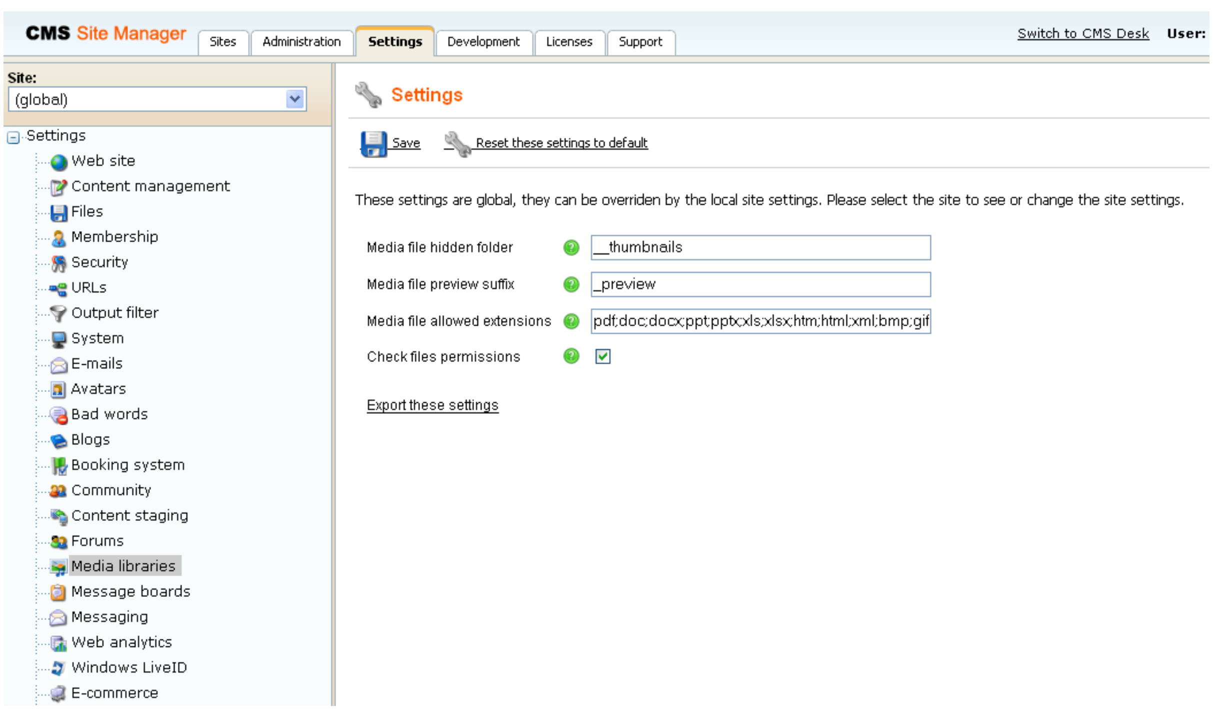
```
this.ltlOutput.Text = "<a href=\"" + url + "\" rel=\"lightbox[gallery]\" title=\"" +
mfi.FileTitle + "\"><img src=\"" + url + "\" border=\"0\"  /></a>";
```

3) Modify the CSS stylesheet for correct displaying of images.

## 45.13 Settings

Settings related to the Media libraries module are located in Site Manager -> Settings -> Files. The following settings are related to the module:

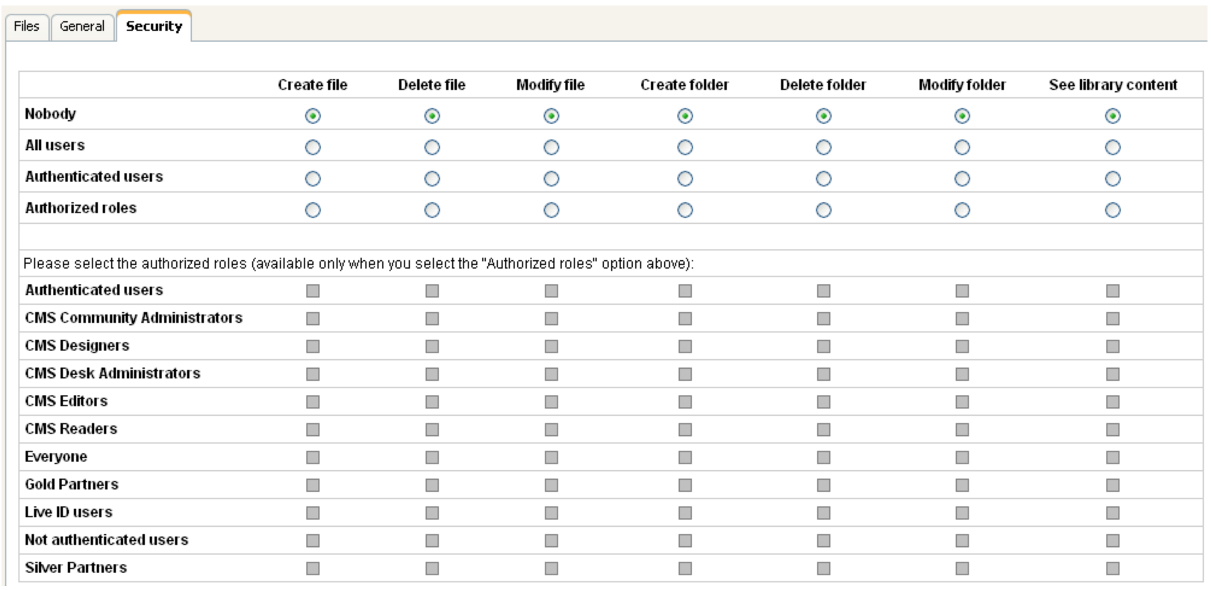
- **Media file preview suffix** - suffix added to preview files; preview files' names are in the following format: < *file name* >_< *file extension* >< *preview suffix* >.< *preview extension* >
- **Media file hidden folder** - name of the folder where thumbnails of media files will be stored; this folder is hidden in the file system by default and thumbnails are generated within it
- **Media file 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



## 45.14 Security

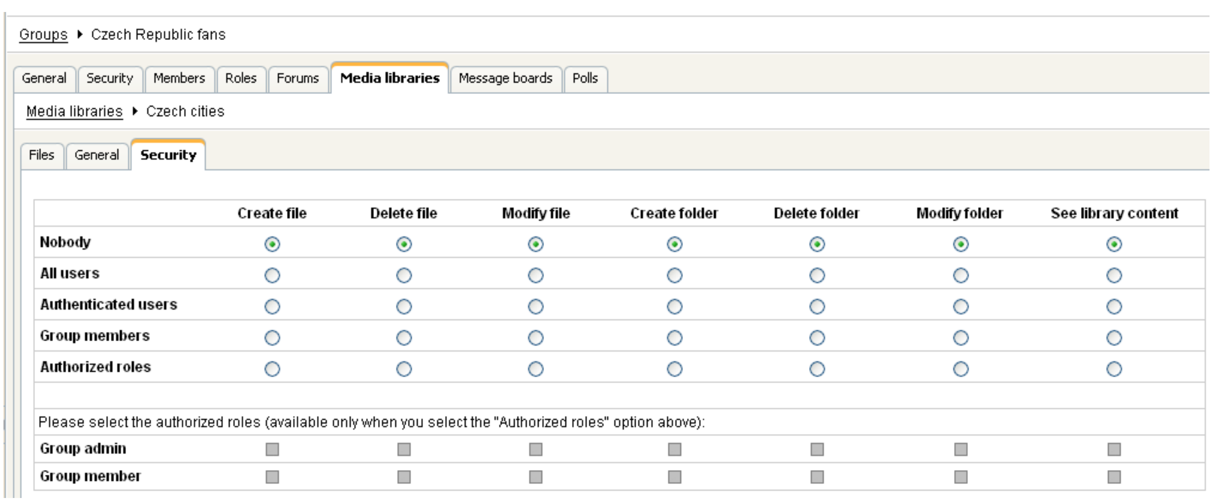
On the Security tab of each media library, you can assign permissions for particular actions to:

- **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 she is the group administrator.

| 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:

- **Display module** - makes the module visible in the administration interface
- **Manage** - allows managing of groups via the administration interface
- **Read** - allows seeing the values in group settings, but does not allow to make any changes to them

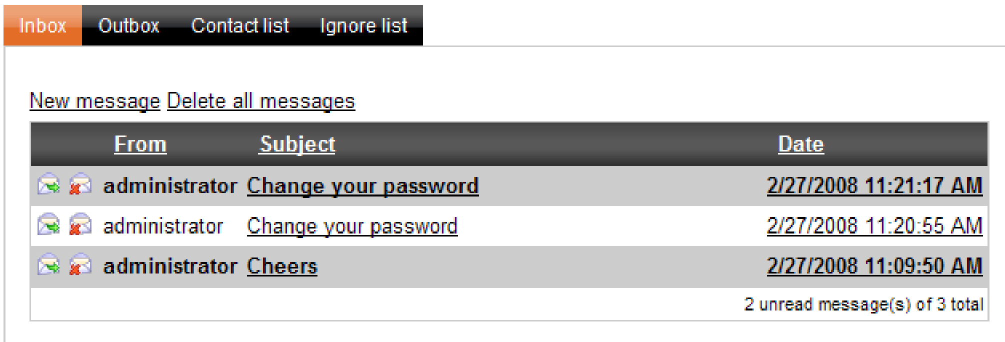
| | Display module | Manage | Read |
|---|---|---|---|
| Authenticated users | ☐ | ☐ | ☐ |
| CMS Community Administrators | ☑ | ☑ | ☑ |
| CMS Designers | ☐ | ☐ | ☐ |
| CMS Desk Administrators | ☑ | ☑ | ☑ |
| CMS Editors | ☑ | ☐ | ☑ |
| CMS Readers | ☑ | ☐ | ☑ |
| Everyone | ☐ | ☐ | ☐ |
| Live ID users | ☐ | ☐ | ☐ |
| Not authenticated users | ☐ | ☐ | ☐ |

# Part XX

# 46 Module Messaging

## 46.1 Messaging module overview

The **Messaging** module allows you to send, receive and manage messages. Its purpose is to provide an internal way of communication with other users of the website.




The **My messages** web part, shown in the screenshot above, is an all-in-one web part consisting of particular messaging web parts found under the **Messaging** folder in the **Web Parts** directory, namely:

- **Contact list:** Allows you to add registered users to your contact list or to remove them from the list.
- **Ignore list:** Allows you to define which users you don't want to receive any messages from.
- **Inbox:** Shows a list of received messages and allows replying to and deleting them.
- **Outbox:** Shows 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 used as a part of the **My messages** web part. You can also choose which of these web parts will be included in the **My messages** web part.

## 46.2 Adding the messaging functionality

Only developers can add the messaging functionality to a website. If you want to do so, switch to **CMS Desk -> Design** and select the page you want to add the functionality to. On the page, click the **Add web part** button in the top right corner of any of the web zones. From the list, select **Messaging** and select the particular web part you want to add. A **Web part properties** window appears, define the specific properties according to which web part you have selected:

### My messages

| | |
|---|---|
| Display inbox | Includes a list of received messages. |
| Display outbox | Includes a list of sent messages. |
| Display contact list | Includes a list of your contacts. |
| Display ignore list | Includes a list of users you don't want to receive messages from. |

### Contact list

| | |
|---|---|
| Page size | Defines the number of contacts per page. |
| No record found text | Defines the text displayed in case there are no contacts in your contact list. |

### Ignore list

| | |
|---|---|
| Page size | Defines the number of contacts per page. |
| No record found text | Defines the text displayed in case there are no contacts in your ignore list. |

### Inbox

| | |
|---|---|
| Page size | Defines the number of messages per page. |
| No record found text | Defines the text displayed in case there are no messages in your inbox. |
| Paste original message | Defines if original message text is inserted in the replying message. |
| Show original message | Defines if original message is displayed when replying to it. |

### Outbox

| | |
|---|---|
| Page size | Defines the number of messages per page. |
| No record found text | Defines the text displayed in case there are no messages in your outbox. |
| Show original message | Defines if original message is displayed. |

### Send message

| | |
|---|---|
| Allow anonymous users | Allows anonymous users to send messages to registered user of the website. |
| Allow anonymous users to select recipient | Allows anonymous users to select the recipient of their message from a list of all registered users. |
| Default recipient of the message | Defines the default recipient of the message. |

Confirm by clicking the **OK** button.

## 46.3 Security

### Anonymous users

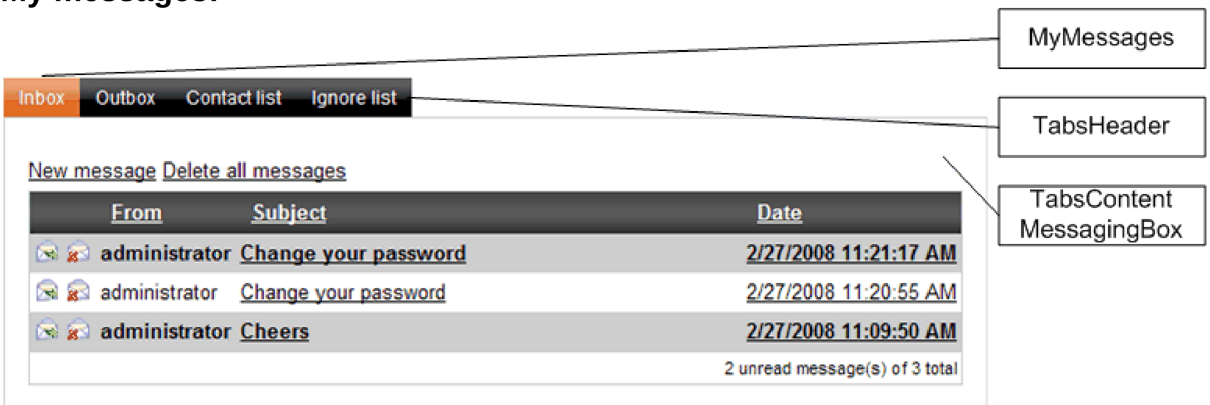
Even unregistered users can send messages to registrated users of your website. This is possible only from the separate **Send message** web part, the **My messages** web part can be used only by registered users.
To allow this functionality, go to **Send message** web part **Properties -> Design** and 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.

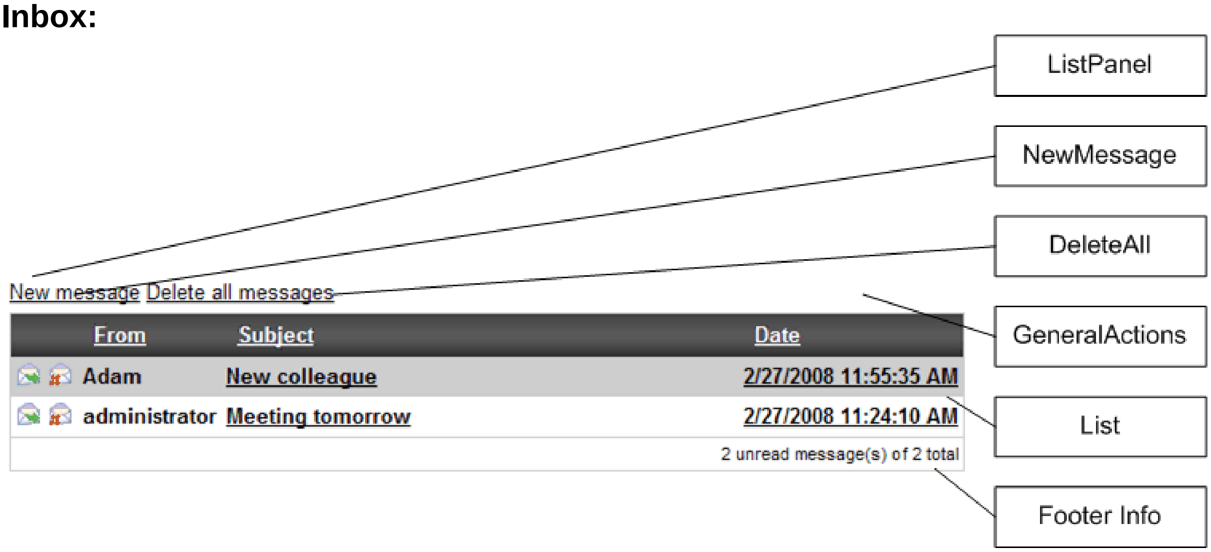
## 46.4 Design and Styles

The design is controlled by CSS styles included in your CSS stylesheet for the given site. The following figures show the messaging web parts' CSS classes:

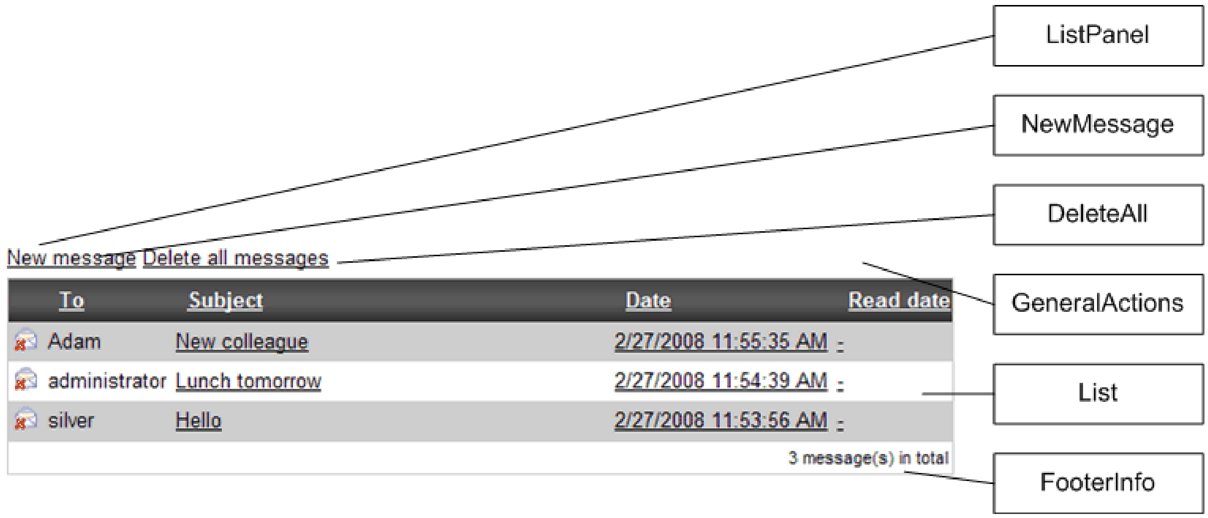
### My messages:



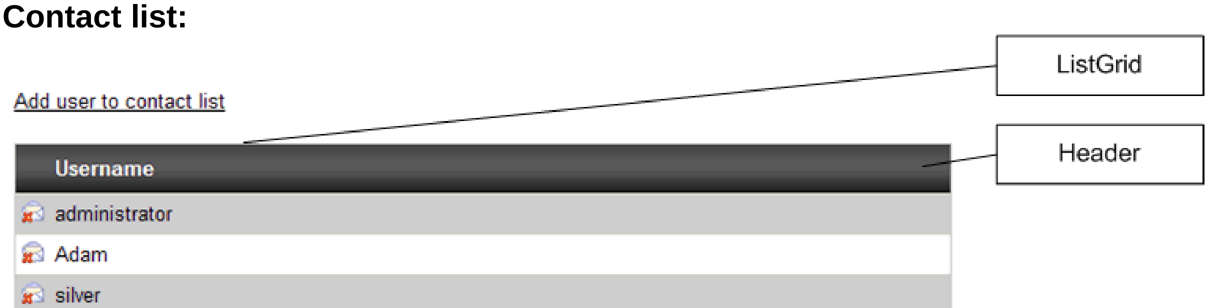

### Inbox:



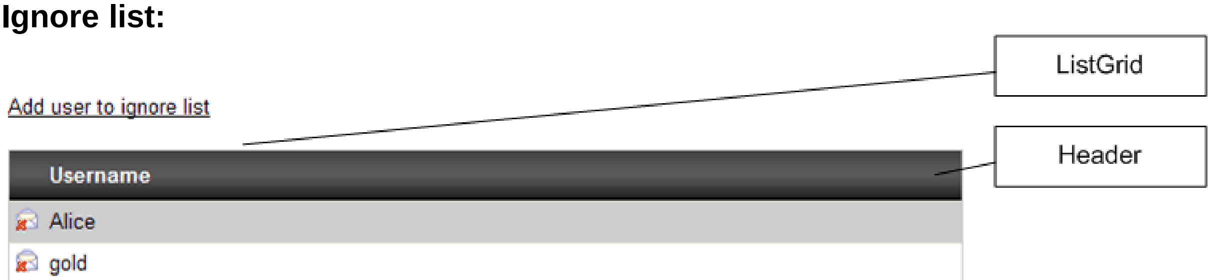
## Outbox:



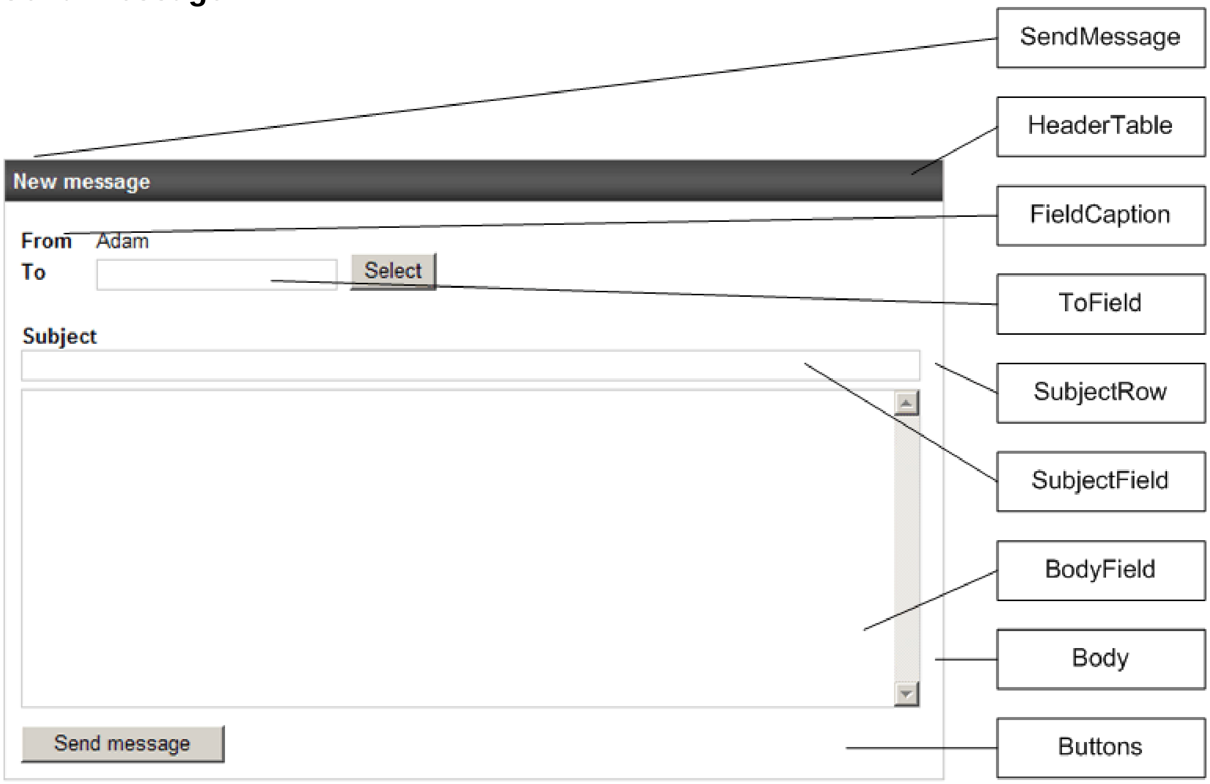

## Contact list:



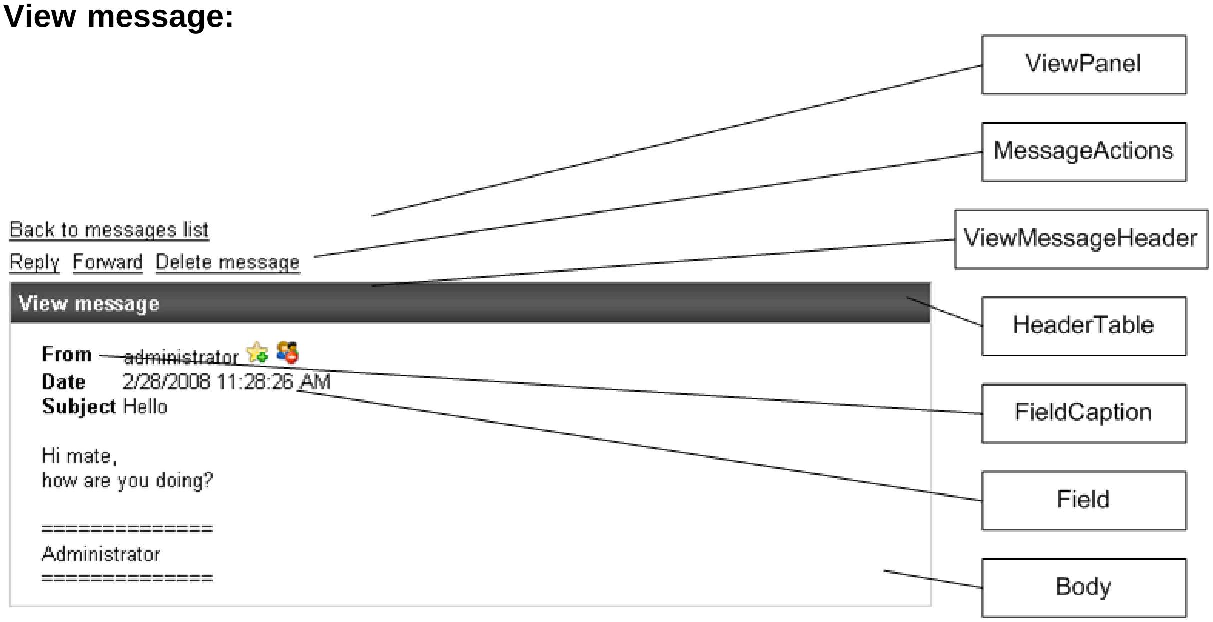

## Ignore list:



## Send message:



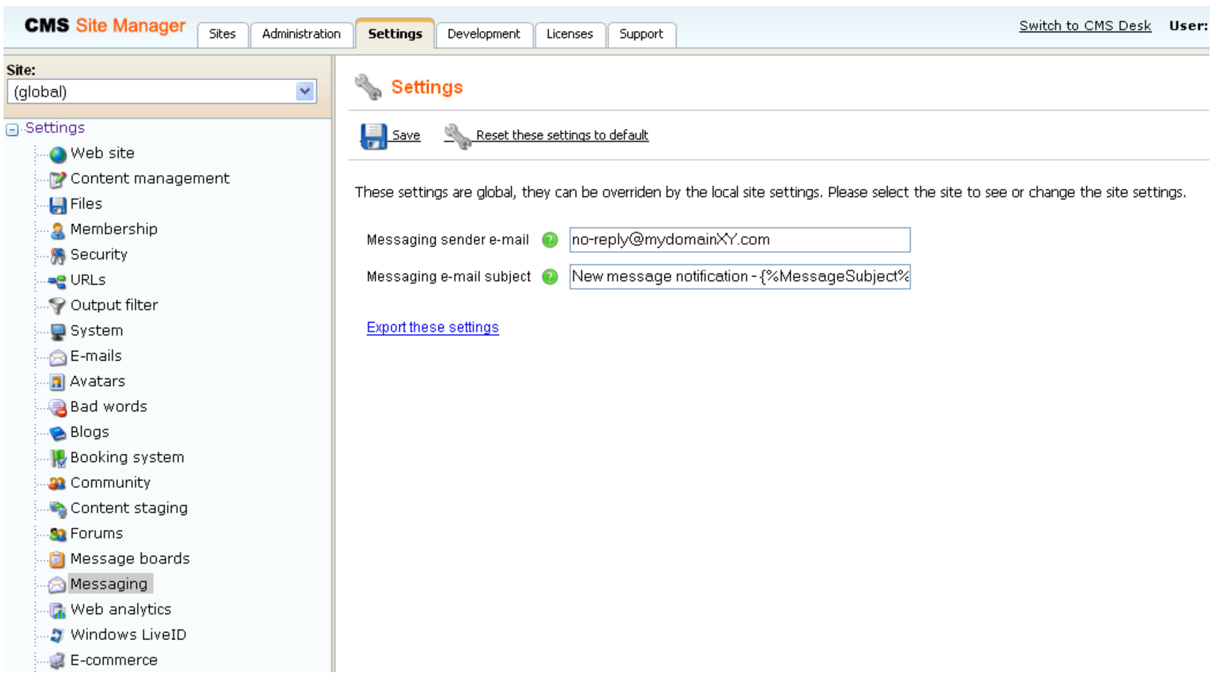

## View message:



## 46.5 Settings

Settings of the Messaging module can be done in **Site manager -> Settings -> Messaging**. The following settings are available:

**Messaging sender e-mail** - e-mail address that will be used a the Sender ('From') address of notification e-mails
**Messaging e-mail subject** - entered text will be used as content of the Subject field of notification e-mails




## 46.6 Messaging API and Internals

### 46.6.1 Database tables and API classes

The Messaging module uses the following database table:

- **Messaging_Message** - this is the only database table used for storing messages

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 these classes.

## 46.6.2 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);
```

## 46.6.3 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);
```

### 46.6.4 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);
```

### 46.6.5 Removing a user form 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);
```

# Part XX

# 47 Module Message boards

## 47.1 Overview

The Message boards module enables site visitors to comment content of a particular page. The module comes with 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 on a particular board, but doesn't allow adding new messages

Each message board is related to a document on that 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.

## 47.2 Using the Message board web part

Message board can be placed on any page (document) on your web site. 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, which can be accessed via  Kentico CMS folder in Windows' Start menu.

- **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

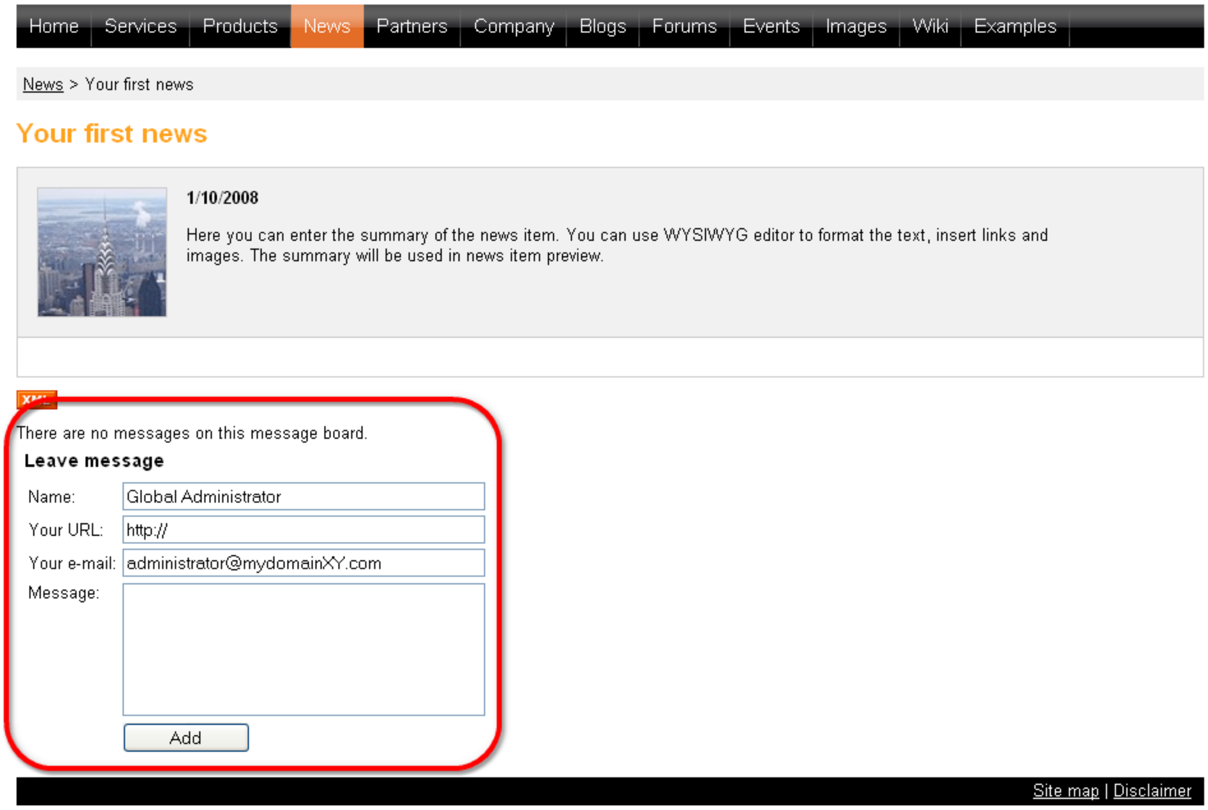
- **Display name** - display name of the message board, it will be used in the administration interface to identify the message board, so it is advisable to use some well-descriptive name under that you will recognize the message board; it is useful to use macros, such as *{%SiteName%} - my first message board*

Click **OK** when you are finished entering the values.

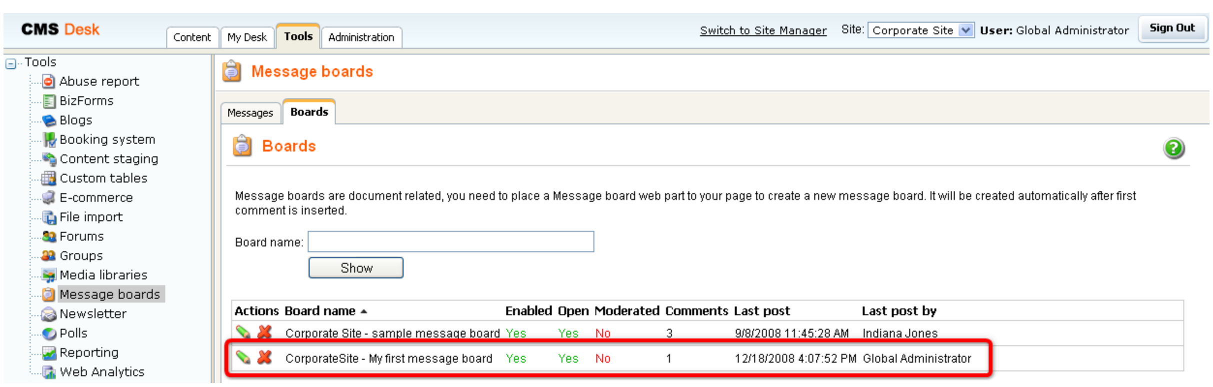
**!! Board default settings !!**

Properties in the **Board default 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 **Live Site** mode now, you should see the for 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.



## 47.3 Administrating message boards

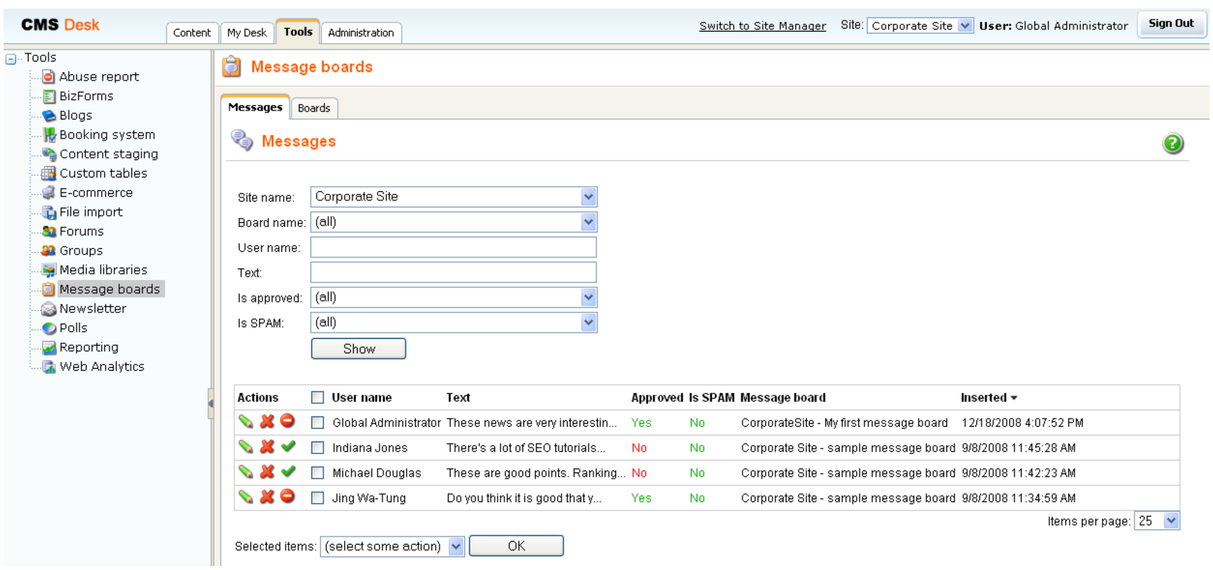
Message boards administration can be performed in **CMS Desk -> Tools -> Message boards**. The section is divided into two tabs.

### Messages tab

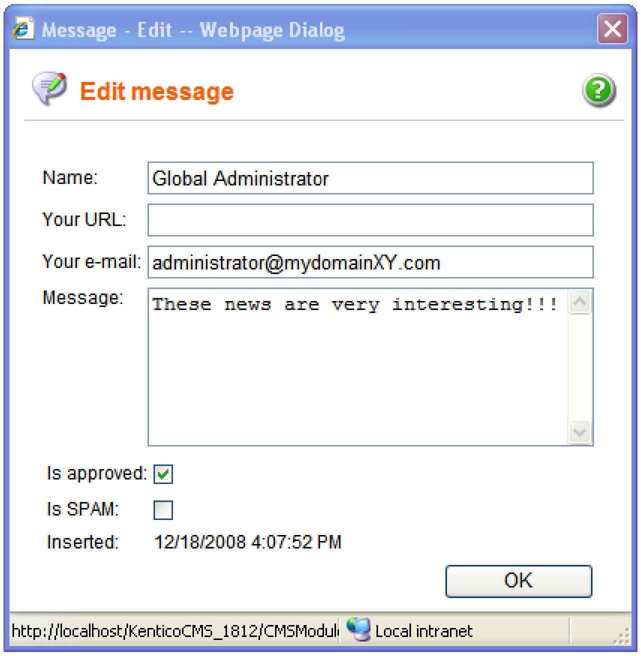
On this tab, moderators can manage board messages. By default, there are only messages requiring approval displayed when you access the page, so that the moderator sees only new messages that need to be approved or rejected. Until the messages aren't approved, they won't be displayed on the message board. Rejected messages won't be displayed too.

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 (not) marked as SPAM



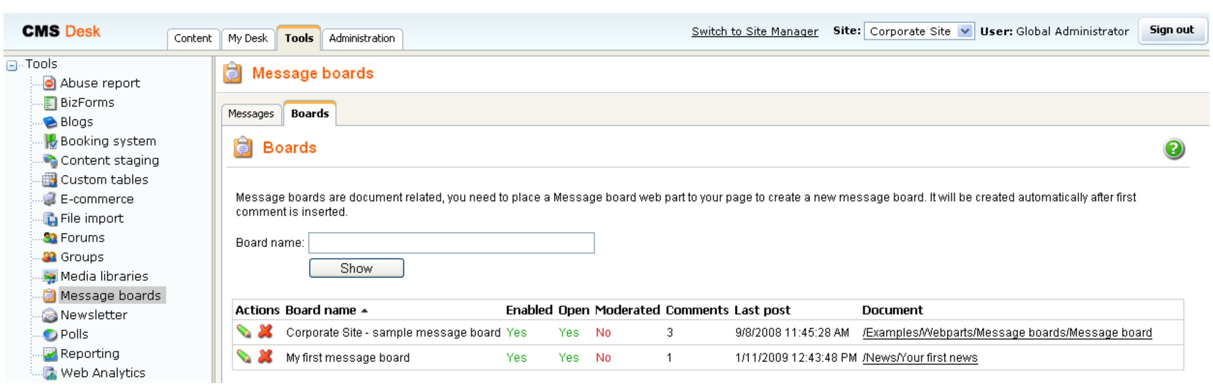
Messages in the list can be **Approved** (), **Rejected** (), **Deleted** () or **Edited** (). If you choose to edit a message, the following window pops-up, letting you make changes to it. You can also select more messages using the check-boxes and perform one of these actions for all of them using 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.



## 47.4 Editing message boards

You can edit message boards' properties **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's properties. When you add the web part to a page, you can set its properties in the web part properties dialog. When 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 effects and you have to make all changes only in this section!

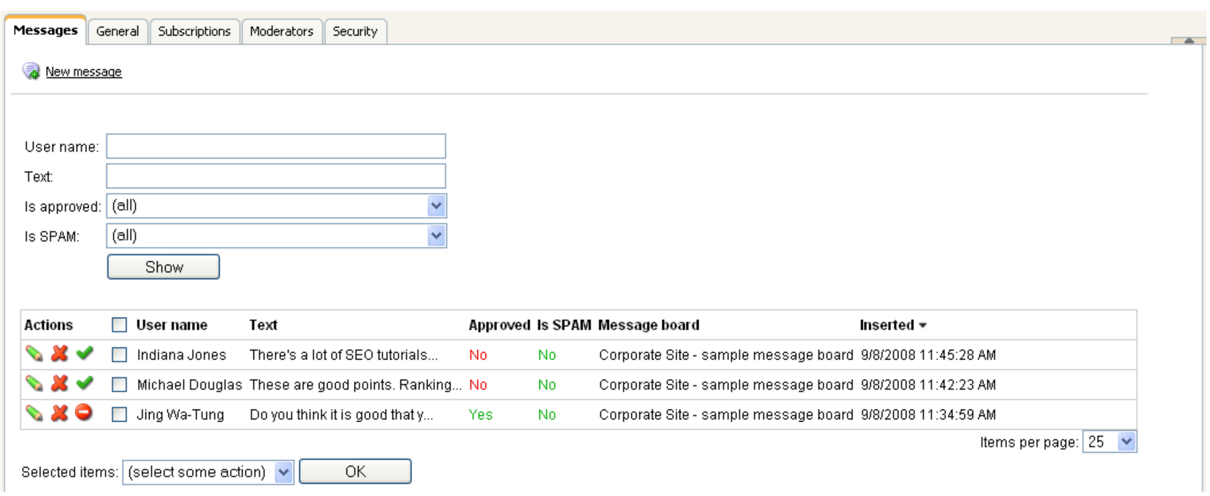
The administration interface for editing message boards is divided into five tabs:

### 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

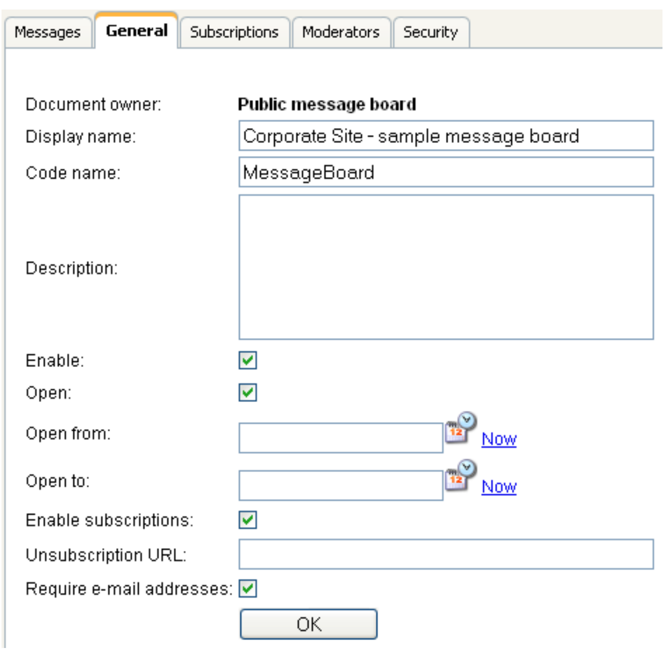
Messages can be **Edited** ( ), **Deleted** ( ), **Approved** ( ) or **Rejected** ( ). You can also select more messages using the check-boxes and perform one of these actions for all of them using the **Selected items** drop-down list and clicking the **OK** button.



## General tab

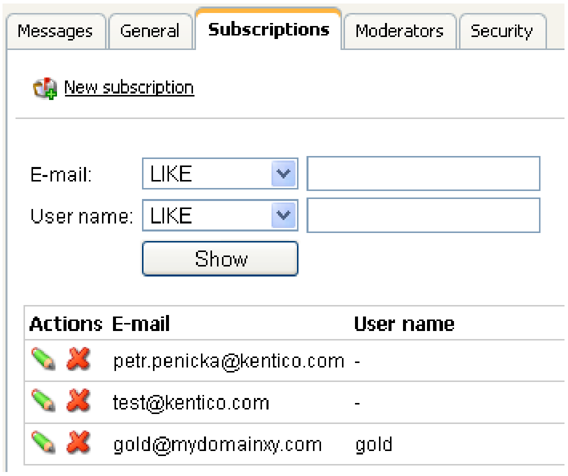
On the general tab, you can specify the following properties of the message board:

- **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 can't 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
- **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 Settings -> Message boards -> Board unsubscription URL will be used
- **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 can not be placed on pages with wildcard URLs
- **Require e-mail address** - if checked, users are required to enter their e-mail address when posting board messages



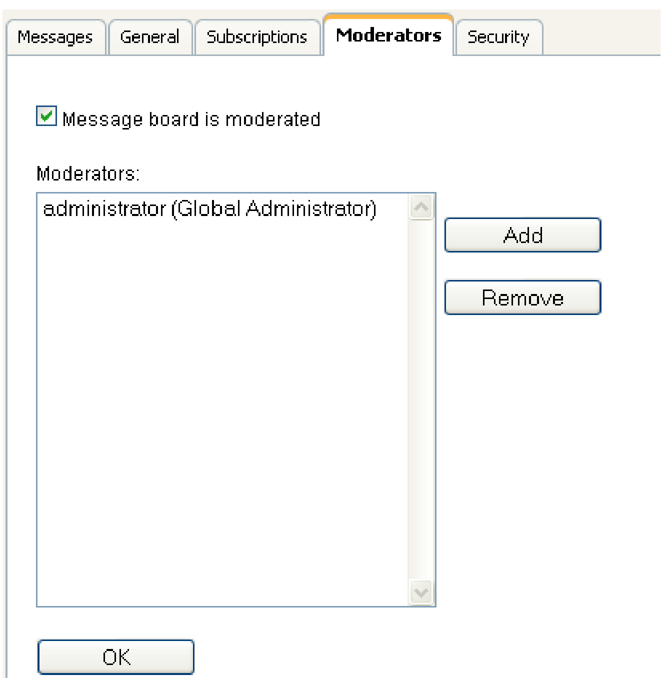
## Subscriptions tab

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.

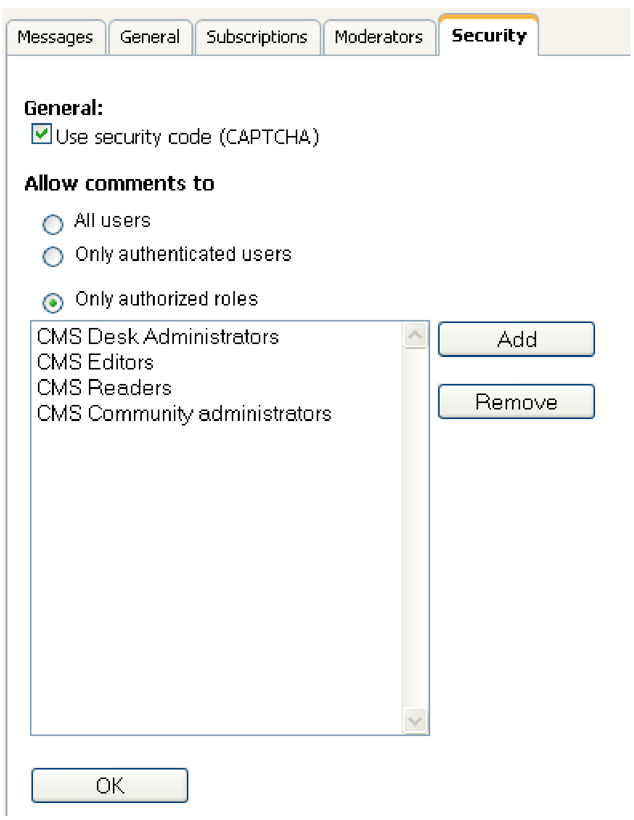
If the **Use security code (CAPTCHA)** check-box is checked, users will have to retype the CAPTCHA security code before adding a new message.

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



## 47.5 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 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.**

As user and group profile pages use Wildcard URLs, 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/{%SiteContet.CurrentUser.UserName%}.aspx
```

You can find a live example of this setting on the sample

2) When you create a **group message board**, which is a board placed on a group's profile, it is recommended to set the the **Board base URL** directly in web part properties, for example like this:

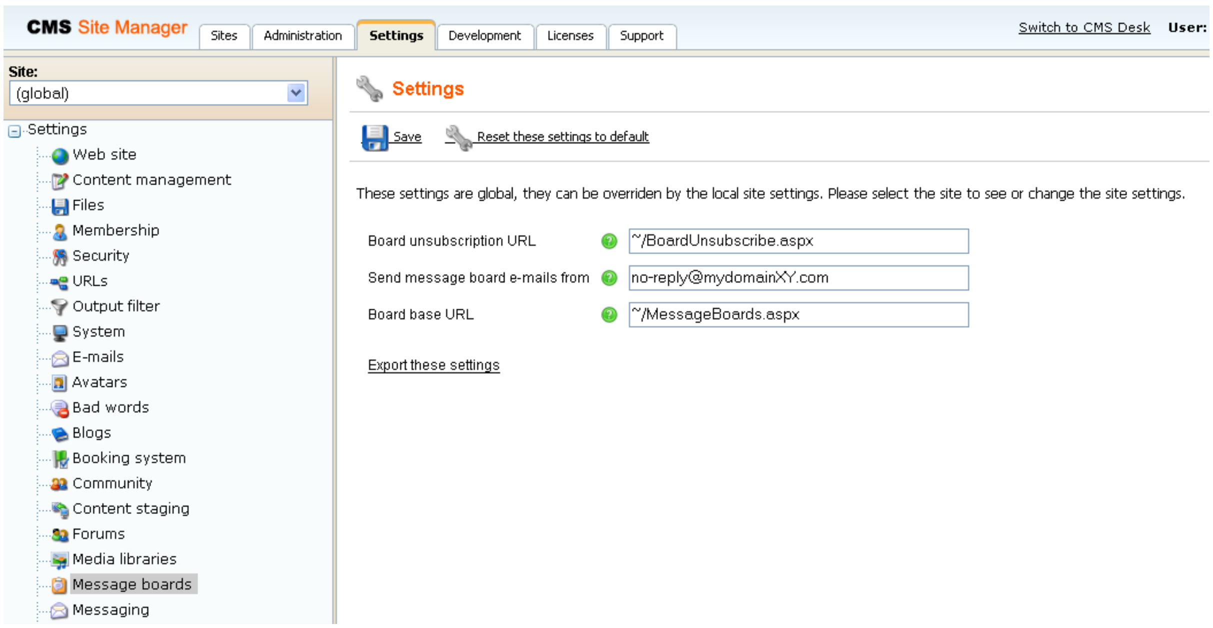
```
~/Groups/{%CommunityContext.CurrentGroup.GroupName%}.aspx
```

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.

## 47.6 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 that 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




## 47.7 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 web part's 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 which users 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. User boards are typically used on user profiles, as you can see on the sample **Community starter site**, on the **Members -> Profile** page.

| Access | Message board owner | Anonymous user | Authenticated user | Authorized role | Owner | Owner in authorized role |
|---|---|---|---|---|---|---|
| All users | Public board | Yes | Yes | Yes | Yes | Yes |
| Authenticated users | Public board | | Yes | Yes | Yes | Yes |
| Authorized roles | Public board | | | Yes | | Yes |
| Owner | Public board | | | | | |
| All users | User | Yes | Yes | Yes | Yes | Yes |
| Authenticated users | User | | Yes | Yes | Yes | Yes |
| Authorized roles | User | | | Yes | | Yes |
| Owner | User | | | | Yes | Yes |

## 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 sample **Community starter site**, 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 |

## 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.

- **Display module** - the module is visible in the administration interface to members of the selected roles
- **Modify** - selected roles' members are allowed to modify the records
- **Read** - selected roles' members are allowed to read the records


| | Display module | Modify | Read |
|---|---|---|---|
| Authenticated users | ☐ | ☐ | ☐ |
| CMS Community administrators | ☑ | ☑ | ☑ |
| CMS Designers | ☐ | ☐ | ☐ |
| CMS Desk administrators | ☑ | ☑ | ☑ |
| CMS Editors | ☑ | ☐ | ☑ |
| CMS Readers | ☑ | ☐ | ☑ |
| Everyone | ☐ | ☐ | ☐ |
| Live ID users | ☐ | ☐ | ☐ |
| Non-authenticated users | ☐ | ☐ | ☐ |

## 47.8 Message board notifications

### 47.8.1 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 are the notifications 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

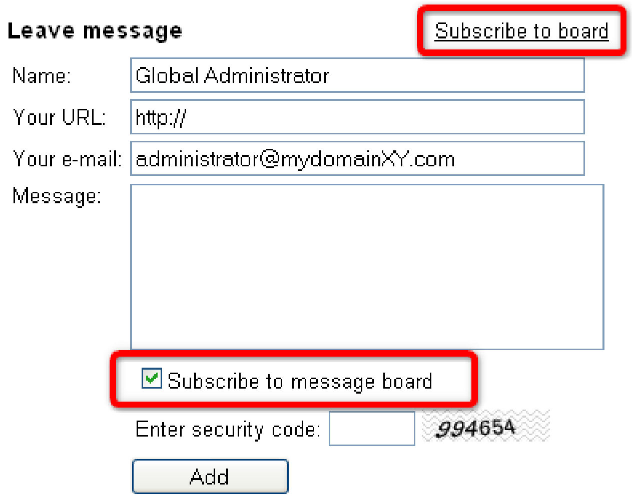
2) other message changes
 - no e-mail is sent

### 47.8.2 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's 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.

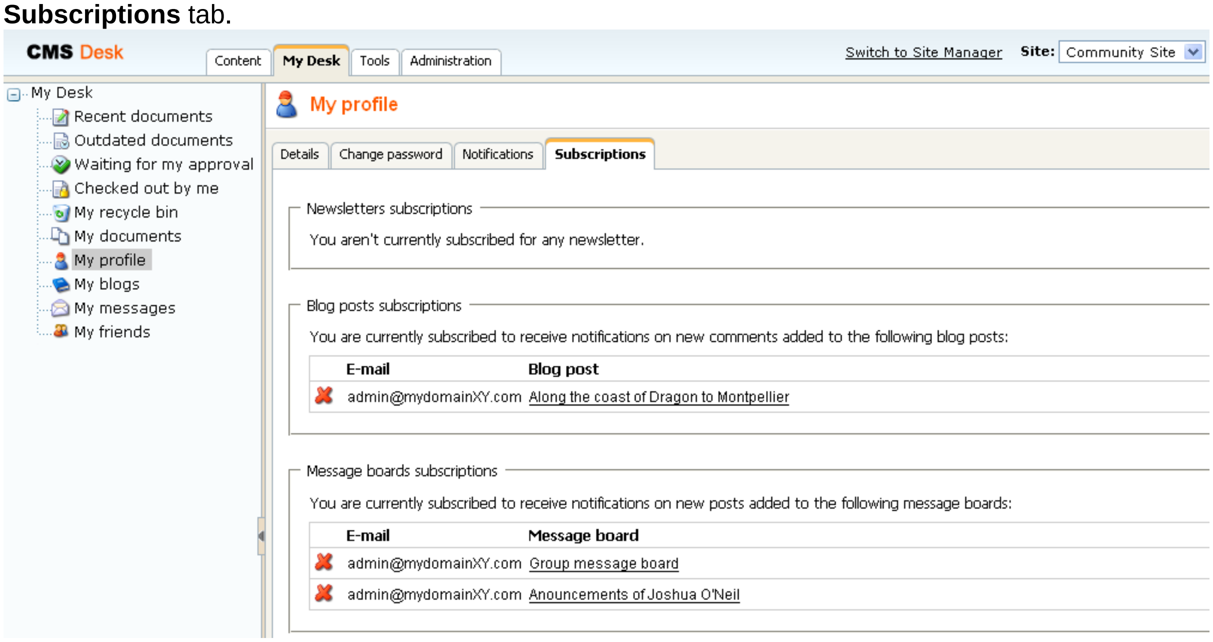
The subscription itself can be done two ways. Users can either check the **Subscribe to message board** check-box, which subscribes them along with adding the message. They can also click the **Subscribe to board** link, which displays only one **E-mail** field. After entering their address and clicking the **Subscribe** button, they can subscribe to notifications about this board without leaving any message.



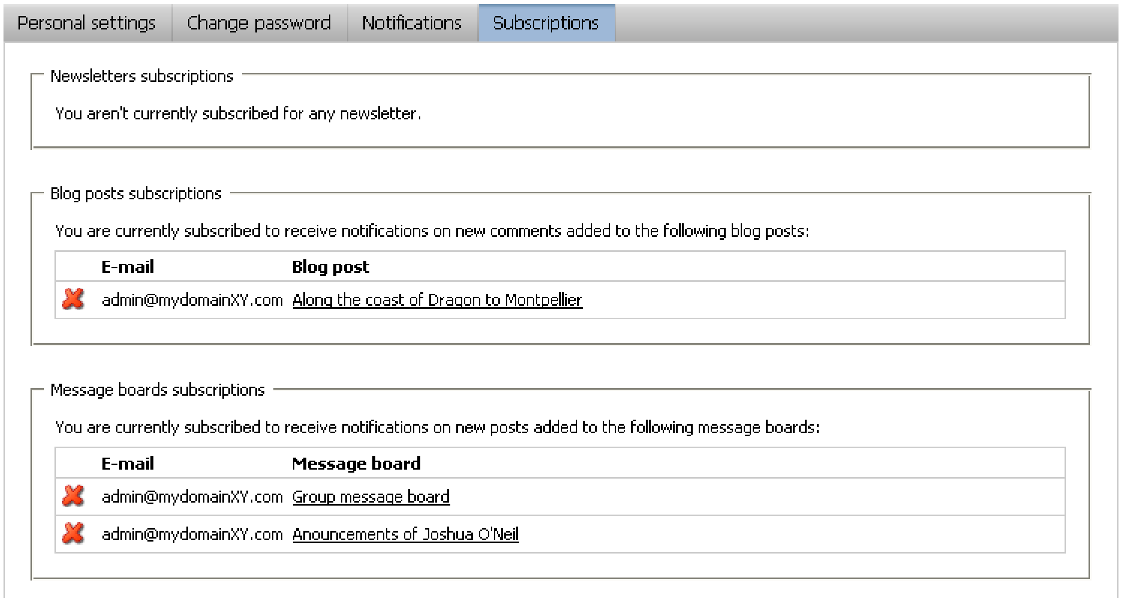

## Subscriptions management

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.

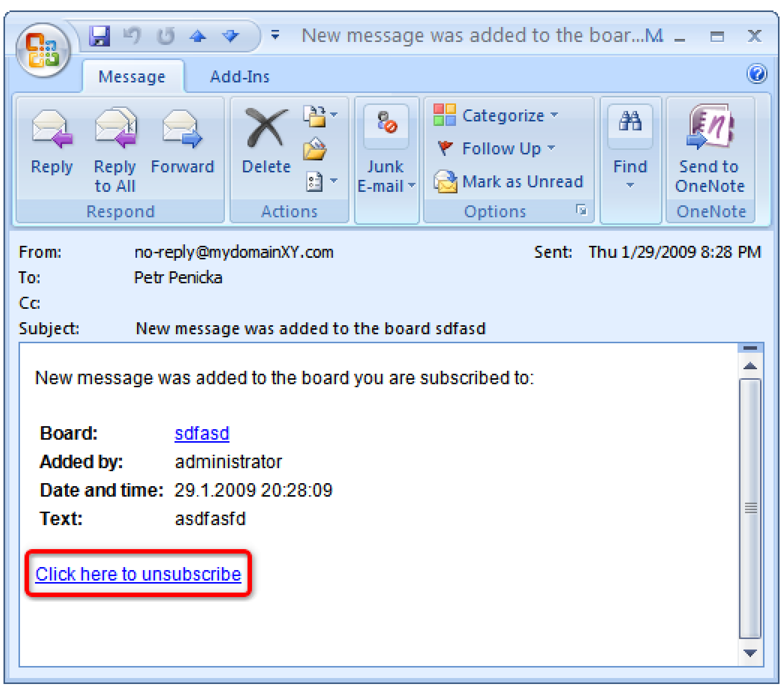


2) On live site, users can view their subscriptions in the **My account** web part. The **Display subscriptions** property of the web part must be enabled for this to be possible.




## 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 some 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 in **CMS Desk -> Tools -> Message boards -> Boards** tab **-> Edit ()the message board -> General** tab.

## 47.8.3 E-mail templates

There are two different e-mail templates that can be used when sending notifications to subscribers and board moderators:

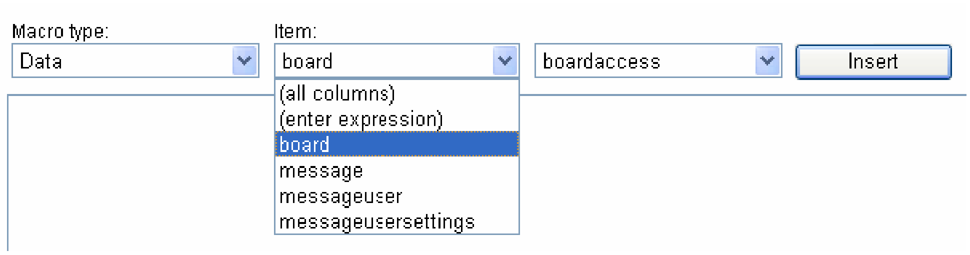
**Subscribers** - e-mails are based on the **Boards - Notification to board subscribers** template
**Moderators** - e-mails are based on the **Boards - Notification to board moderators** template

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

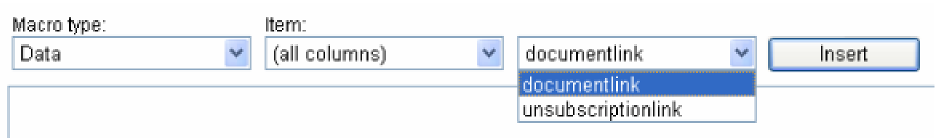
**Example**: {%Message.MessageEmail%}



## Source macros

**DocumentLink** - link to the document where the board is placed
**UnsubscriptionLink** - unsubscription link

**Example**:: {%DocumentLink %}



# Part XX

# 48 Module Newsletters

## 48.1 Newsletter module overview

The Newsletter module allows you to author and mail out e-mail newsletters. The newsletters can be of two types:

- Static newsletters - you edit and send out every issue manually. The newsletters are based on predefined templates.
- Dynamic newsletters - the e-mails are sent out to all subscribers on a specified interval. The content is dynamically taken from specified page.

Newsletters can be managed in **CMS Desk -> Tools -> Newsletter** section.

## 48.2 Creating a static newsletter

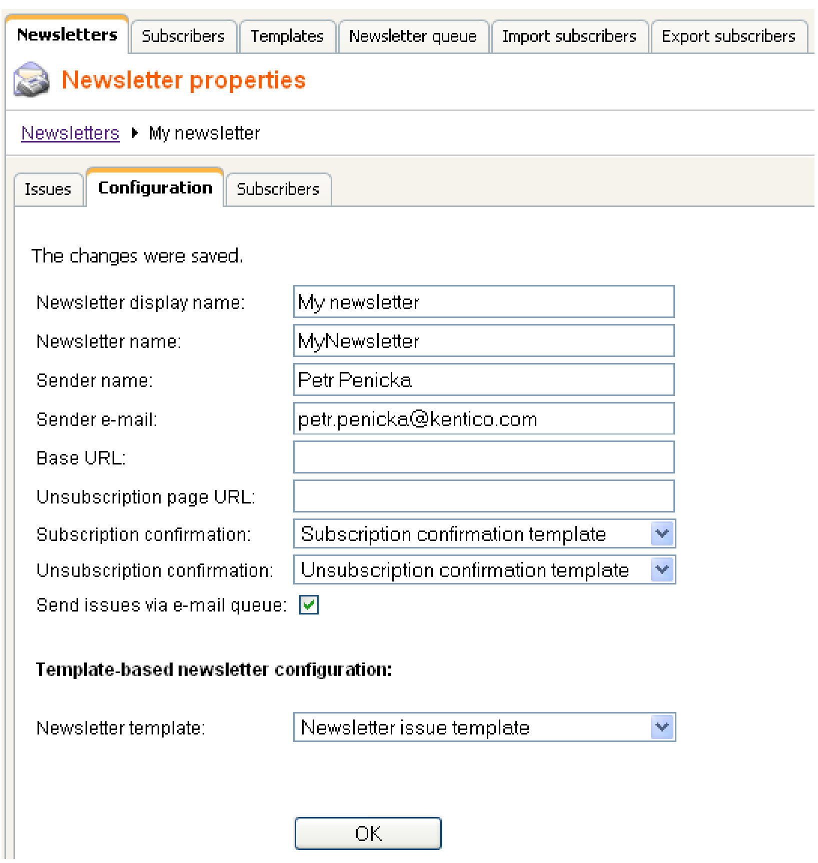
In this chapter, you will learn how to create a static newsletter and author your first newsletter issue.

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

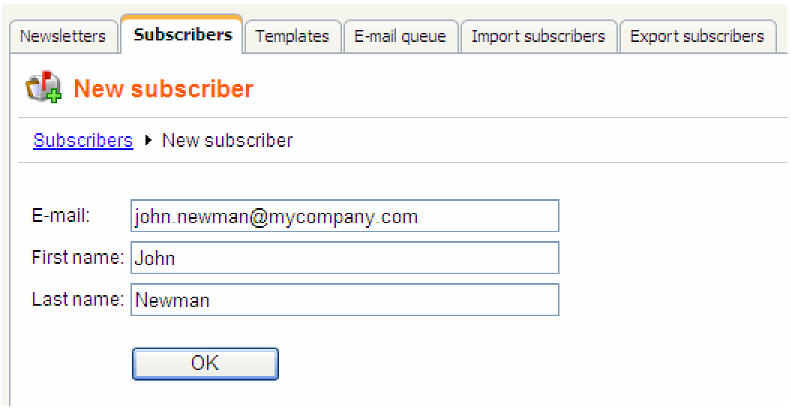
Choose **Template based newsletter** and use the **Newsletter issue template.** Click **OK**. The following three options can now be entered additionally:

- **Base URL:** the base URL of your web site 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 web site. Example: https://www.mydomain.com
- **Unsubscription page URL:** ~/SpecialPages/Unsubscribe.aspx (the page with Newsletter unsubscription web part)
- **Send issues via e-mail queue** - if checked, newsletter issues will be sent via the e-mail queue



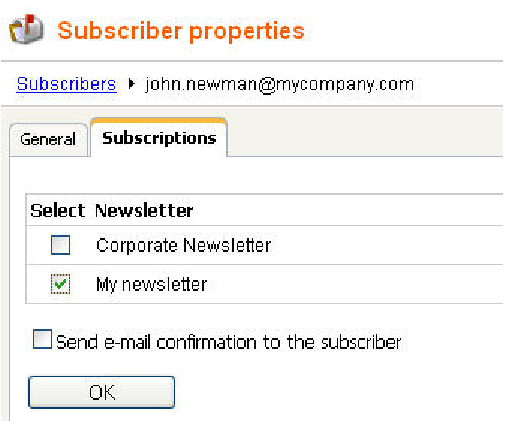
Click **OK**. We will create a new subscriber now. Go to the main **Subscribers** tab (in the top tab menu) and click **New subscriber**. Enter the following details:

- **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.

Go to the **Subscriptions** tab, check the **My newsletter** check-box and click **OK**.




If you check the **Send e-mail confirmation to the subscriber** check-box, the subscriber will also receive an e-mail confirmation of her subscription.

Your newsletter is configured. Now you can create new issues.

## Authoring newsletter issues

Go to **Newsletters** and edit **My newsletter**. Click **New issue** on the **Issues** tab. The wizard will guide you through the process of creating a new newsletter issue.

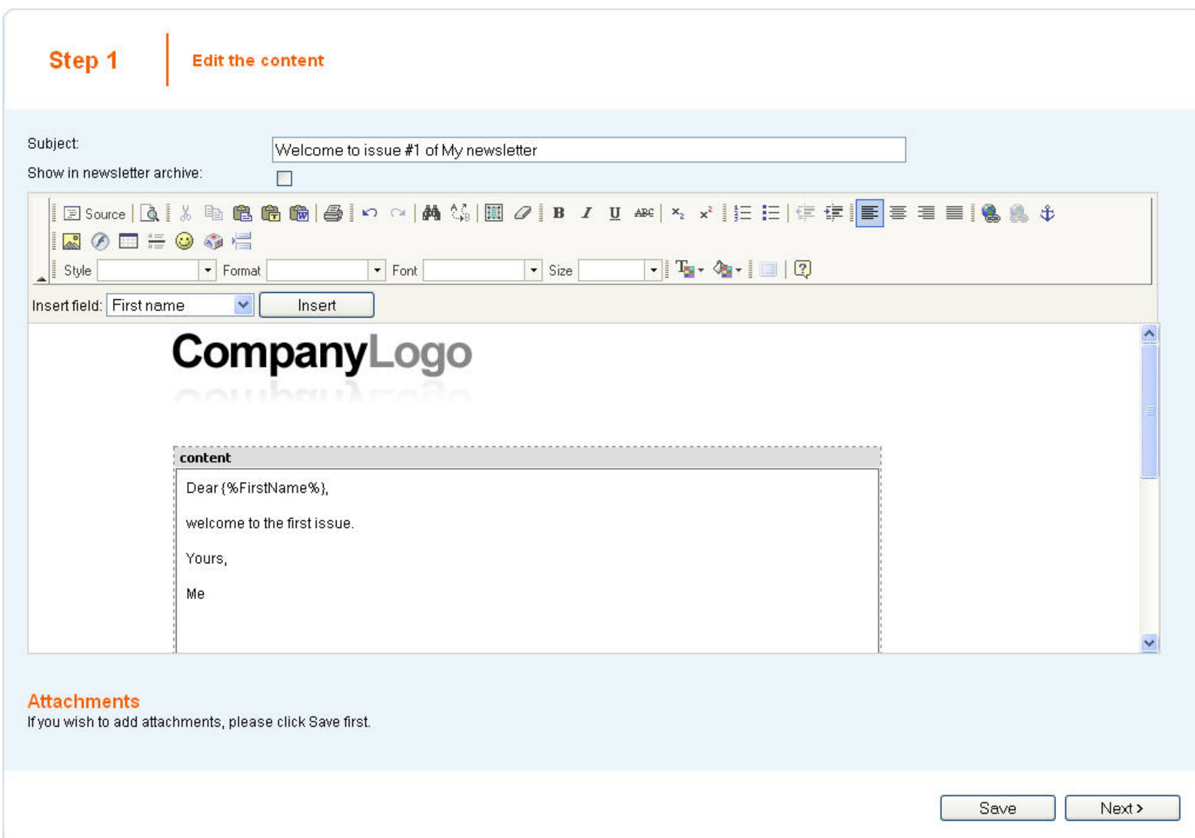
Enter the following **subject**: Welcome to issue #1 of My newsletter

Enter the following text into the **body**:

> Dear ,
>
> welcome to the first issue.
>
> Yours,
>
> Me

place the cursor after the Dear word, choose **First name** value in the **Insert field** drop-down list and click the **Insert** button. The macro text {%FirstName%} is placed into the text. This macro is automatically replaced with subscriber's first name during the e-mail merge.

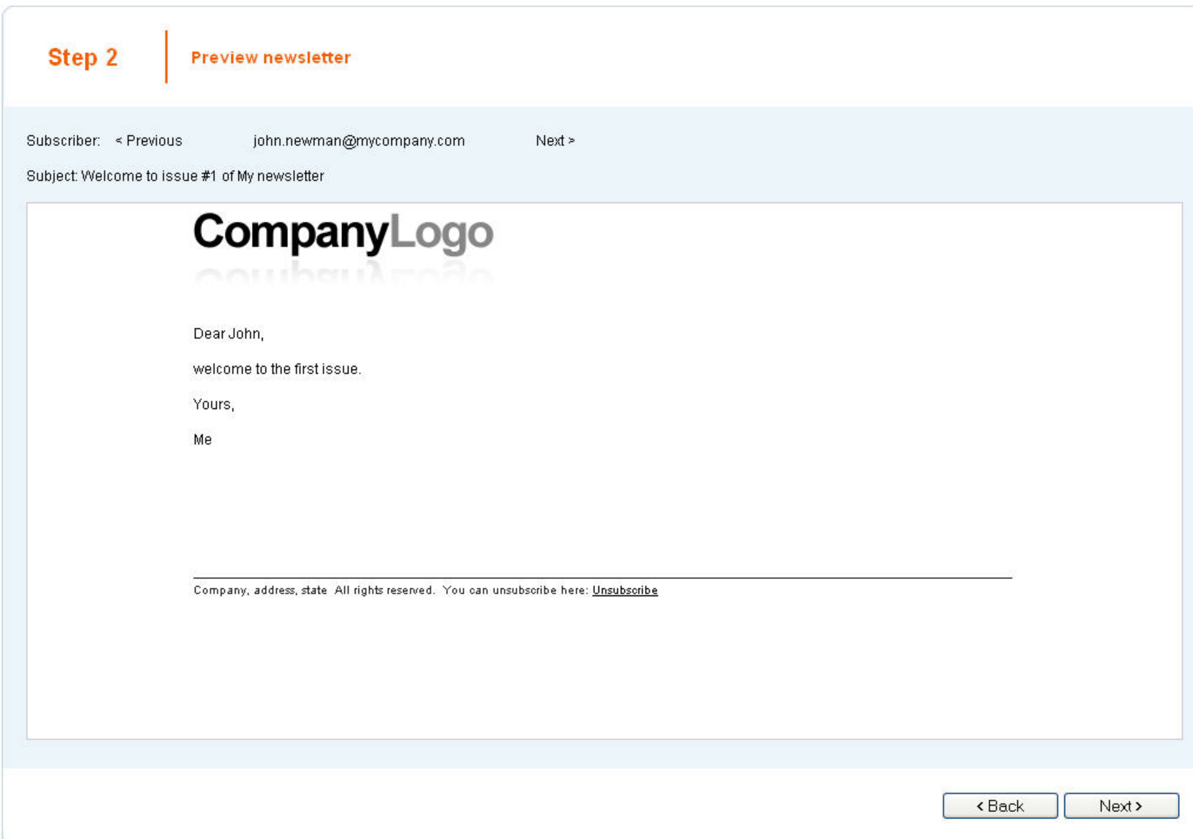
The issue now looks like this:



**Newsletter templates**

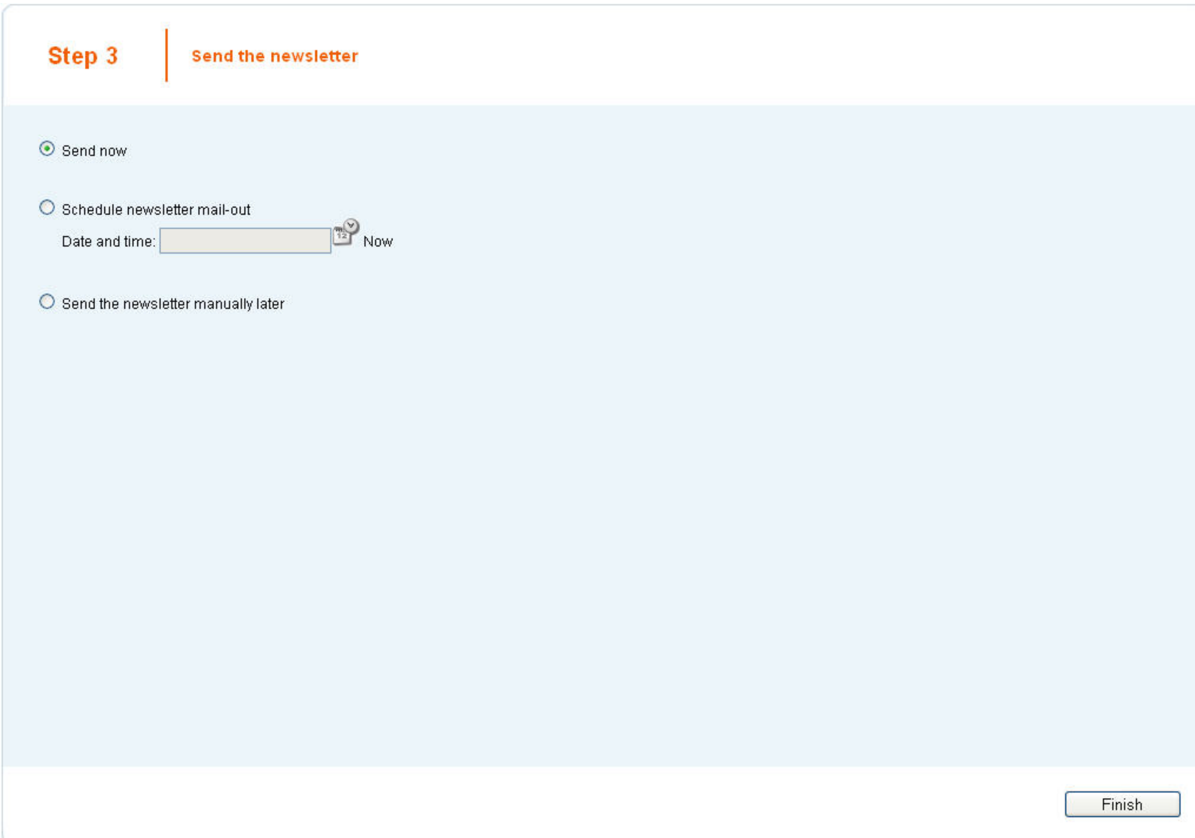
The structure of the newsletter text is defined by the newsletter issue template that can be edited in the **Templates** section. Templates are described in chapter Newsletter templates.

Click **Next**. Now you can preview the newsletter text for each subscriber. You can click **Back** and modify the text.

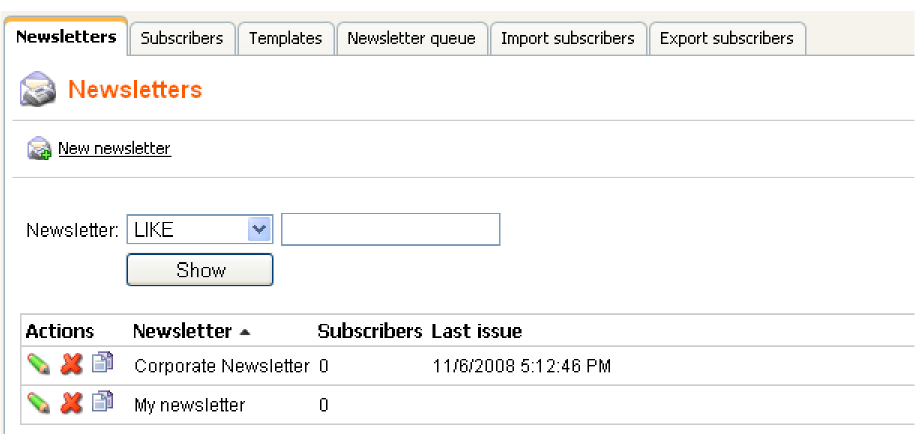


Click **Next**. Now you can choose when the newsletter issue should be sent out:

- **Send now** - the newsletter issue is sent out immediately to all subscribers
- **Scheduled newsletter mail-out** - the newsletter issue is sent out on 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 been already sent out and how many subscribers unsubscribed after receiving this issue:



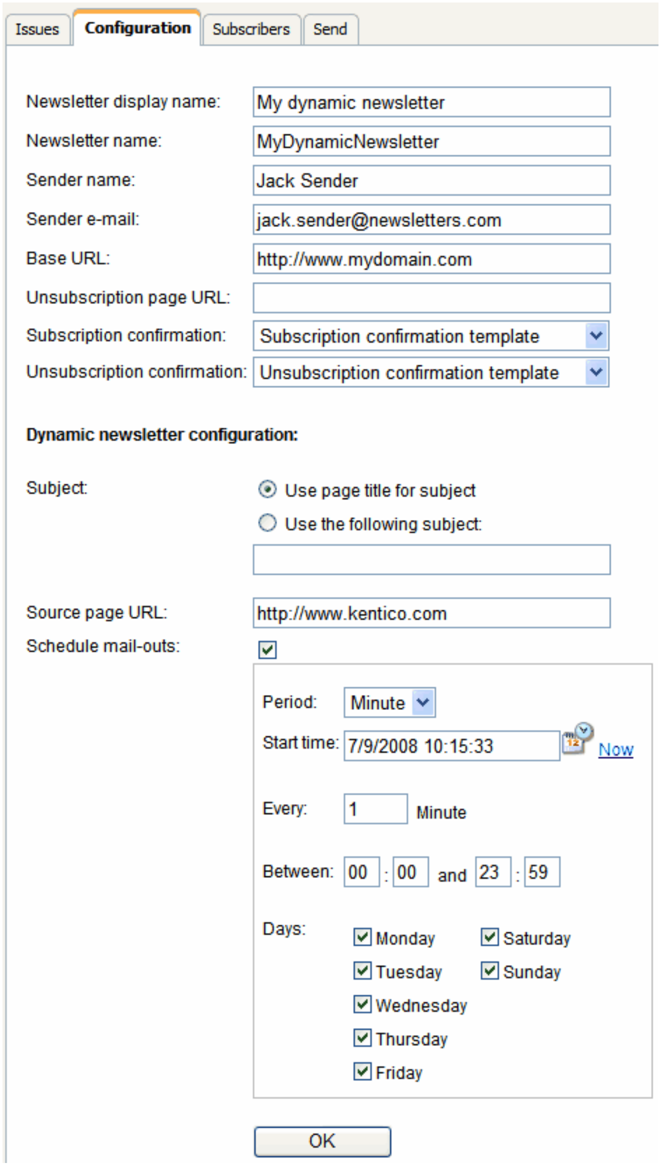

If you do not receive the e-mail, please follow the instructions in chapter Troubleshooting the problems with e-mails.

## 48.3 Creating a dynamic newsletter

Dynamic newsletters contain content of the given page and they are sent out automatically on a regular basis, using the built-in scheduling system.

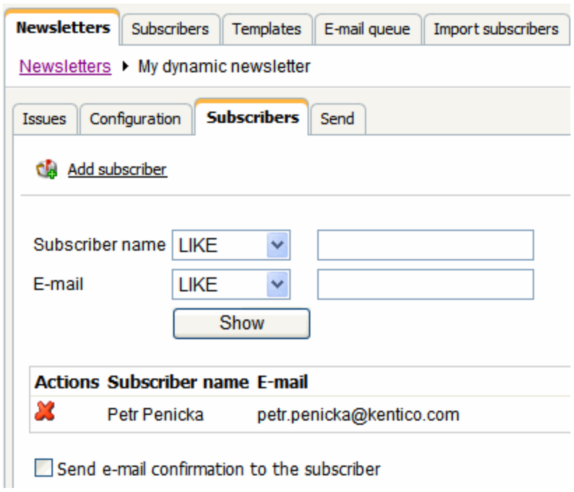
Go to **CMS Desk -> Tools -> Newsletter** and click **New newsletter**. Enter the following details:

- **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
- **Base URL**: the base URL of your web site 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 web site. Example: https://www.mydomain.com
- choose **Dynamic newsletter** and enter the following details:
  - **Source page URL**: http://www.google.com
  - **Schedule mail-outs**:
    - **Period**: Minute
    - **Start time**: use the date-time picker to select the current date and time (Now)
    - **Every**: 1 minute
    - **Between**: 00:00 and 23:59
    - **Days**: check all days

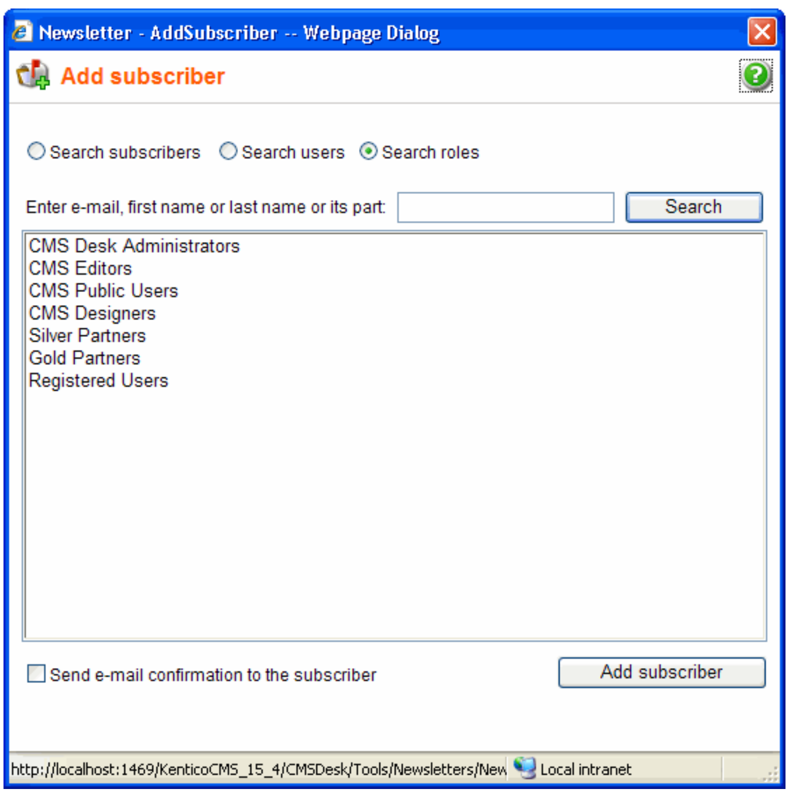

Issues
Configuration
Subscribers
Send
Newsletter display name:
My dynamic newsletter
Newsletter name:
MyDynamicNewsletter
Sender name:
Jack Sender
Sender e-mail:
jack.sender@newsletters.com
Base URL:
http://www.mydomain.com
Unsubscription page URL:
Subscription confirmation:
Subscription confirmation template
Unsubscription confirmation:
Unsubscription confirmation template
Dynamic newsletter configuration:
Subject:
Use page title for subject
Use the following subject:
Source page URL:
http://www.kentico.com
Schedule mail-outs:
Period:
Minute
Start time:
7/9/2008 10:15:33
Now
Every:
1
Minute
Between:
00
00
and
23
59
Days:
Monday
Saturday
Tuesday
Sunday
Wednesday
Thursday
Friday
OK

Click **OK**.

Go to the **Subscribers** tab for this newsletter and add the previously created subscriber to this newsletter.

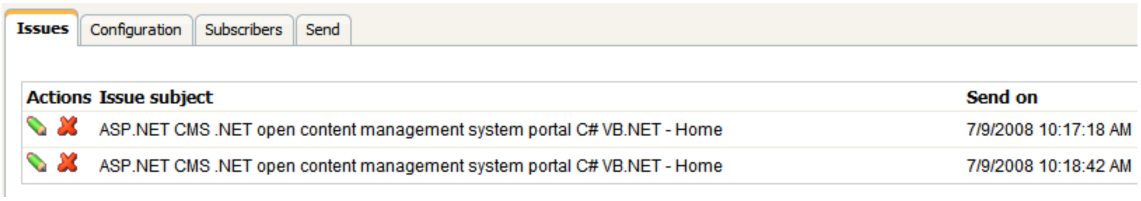



To do this, click the **Add subscriber** link, the following window appears:

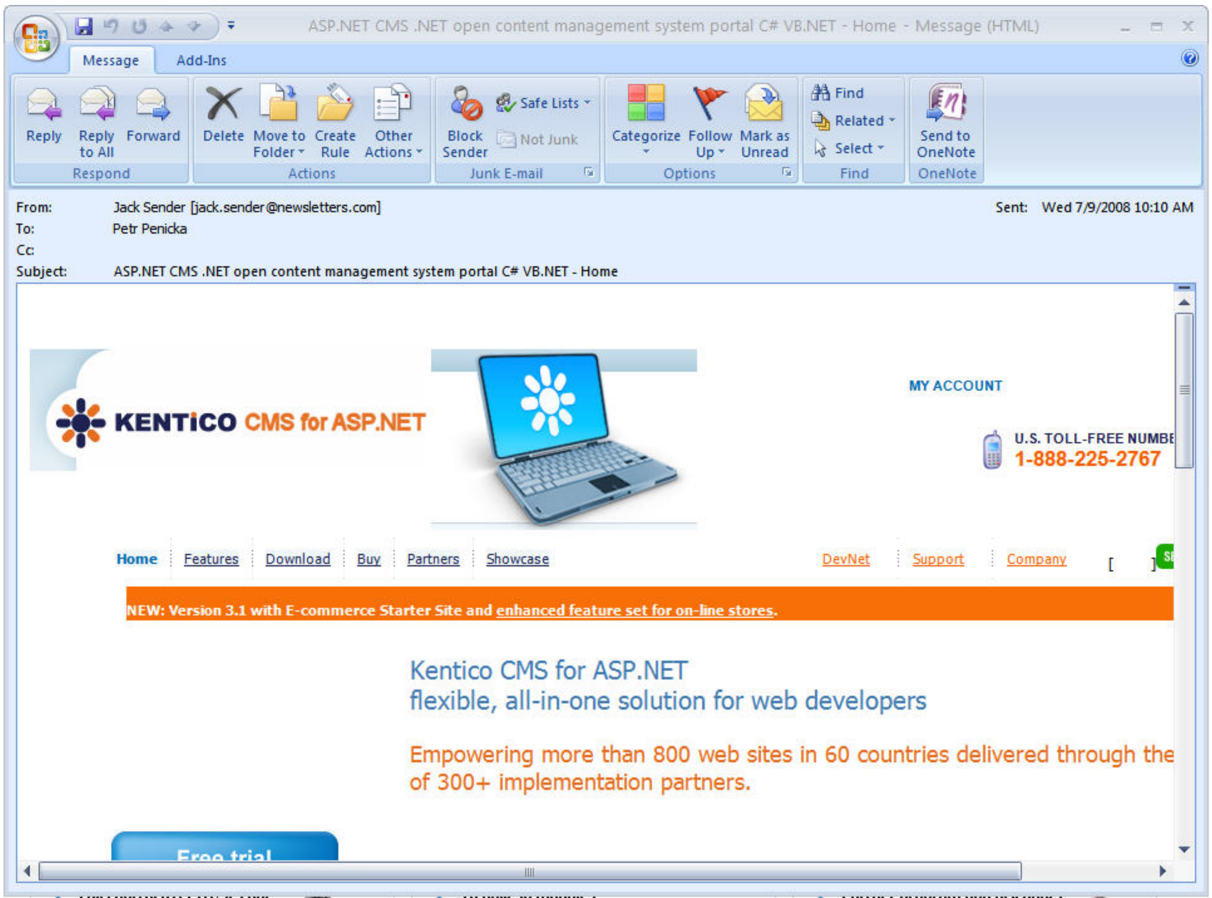



The three radio buttons at the top of the window are used for selecting the searched category. You can search among subscribers of all newsletters, users of the website or among user roles. Make the selection, enter the text you are searching for into the text field and click the **Search** button. The searched text can be an e-mail address, a name or its part. If no search text is entered, a list of all entries in the selected category will be displayed. Select the subscribers you want to add and click the **Add subscriber** button. 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.

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.




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 web site. You can create a new page only for the newsletter purposes that will display e.g. new articles added to your web site during the last month.

### Blocking the dynamic newsletter mail-out

If you want to block the mail-out of the page (e.g. if there are no new articles), set the page title to ##DONOTSEND## and the newsletter will not be sent.

## 48.4 Integrating newsletters into the site

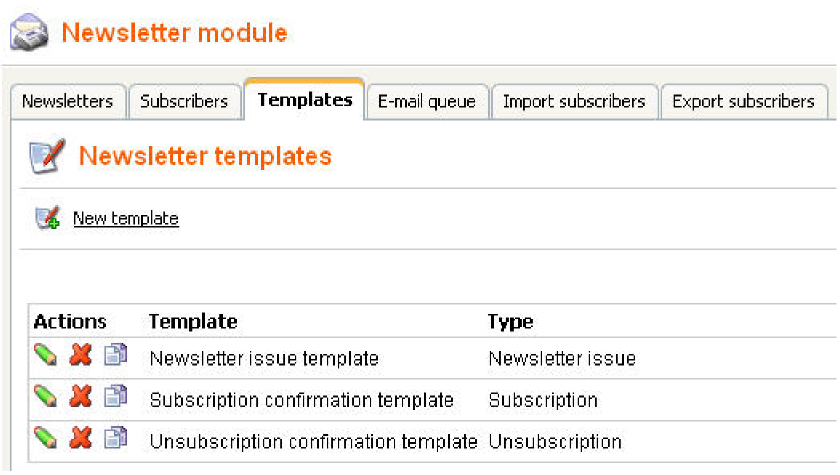
You can integrate the newsletter into your web site using two web parts:

- **Newsletter subscription** - displays newsletter subscription dialog
- **Newsletter unsubscription** - unsubscribes the user and displays a confirmation message when the user clicks the "**Unsubscribe**" link in the newsletter issue
- **Unsubscription request** - this web part allows newsletter subscribers to request unsubscription e-mail by submiting their e-mail address

You can find more details on the properties of these web parts in **Kentico CMS Web Parts and Controls Reference**.

## 48.5 Newsletter templates

The e-mails sent by the Newsletter module can be modified using the templates. The templates can be managed in **CMS Desk -> Tools -> Newsletter -> Templates**.




There are three types of templates:

- **Newsletter issue** - this is template that defines the layout and design of newsletter issues. It contains editable regions where newsletter authors can enter the content.
- **Subscription** - e-mail message sent when the user subscribes to the newsletter.
- **Unsubscription** - e-mail message sent when the user unsubscribes.

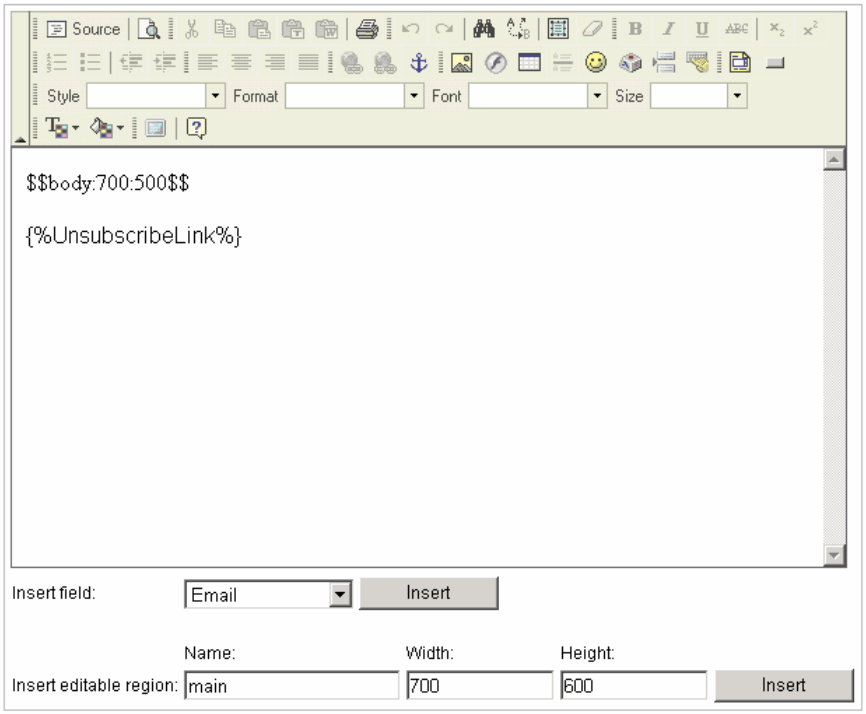
All types of newsletter templates can contain the following macro expressions anywhere in the text:

- {%Email%}
- {%FirstName%}
- {%LastName%}
- {%UnsubscribeLink%}

## Editing newsletter issue templates

The issue templates consist of:

- **Header** (the leading HTML code including the <html> element - here you can place your CSS styles.
- **Body** - the layout of the e-mail - here you can place static text, macros and editable regions. The editable regions allow users to enter 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** - the closing HTML code.
- **CSS stylesheet** - the CSS styles used for the newsletter; these styles are used by the newsletter issue editor and they are included in the e-mails.

# 48.6 Troubleshooting

### Problems with e-mails

If you do not receive 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 in **CMS Desk -> Administration -> Scheduled tasks**.

2. Go to **CMS Desk -> Tools -> Newsletter -> Newsletter queue**. If some e-mail failed, you will find here the exception. After you resolve the technical issue, you can resend all failed e-mails by clicking **Resend all failed**.

3. Make sure you're using correct e-mail addresses.

4. Make sure the newsletter issue is not blocked by some antispam software.

5. 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 chapter SMTP server configuration. You can also test e-mail settings in **Site Manager -> Administration -> System -> E-mail**.

Newsletter e-mails 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. The first key enables logging of all sent e-mails to *<web root>/AppData/logemails.log*. The second key causes that all sent e-mails will be logged, but not actually sent. This is helpful when you need to test the functionality but do not want the testing e-mails to be really sent.

```
<add key="CMSLogEmails" value="true"/>
<add key="CMSDebugEmails" value="true"/>
```

### Problems with unsubscription links

If you encounter problems with unsubscription link resolving, you should set the **Base URL** property of the newsletter. This property can be set in **CMS Desk -> Tools -> Newsletter**, if you choose to **Edit** () the newsletter and switch to its **Configuration** tab.

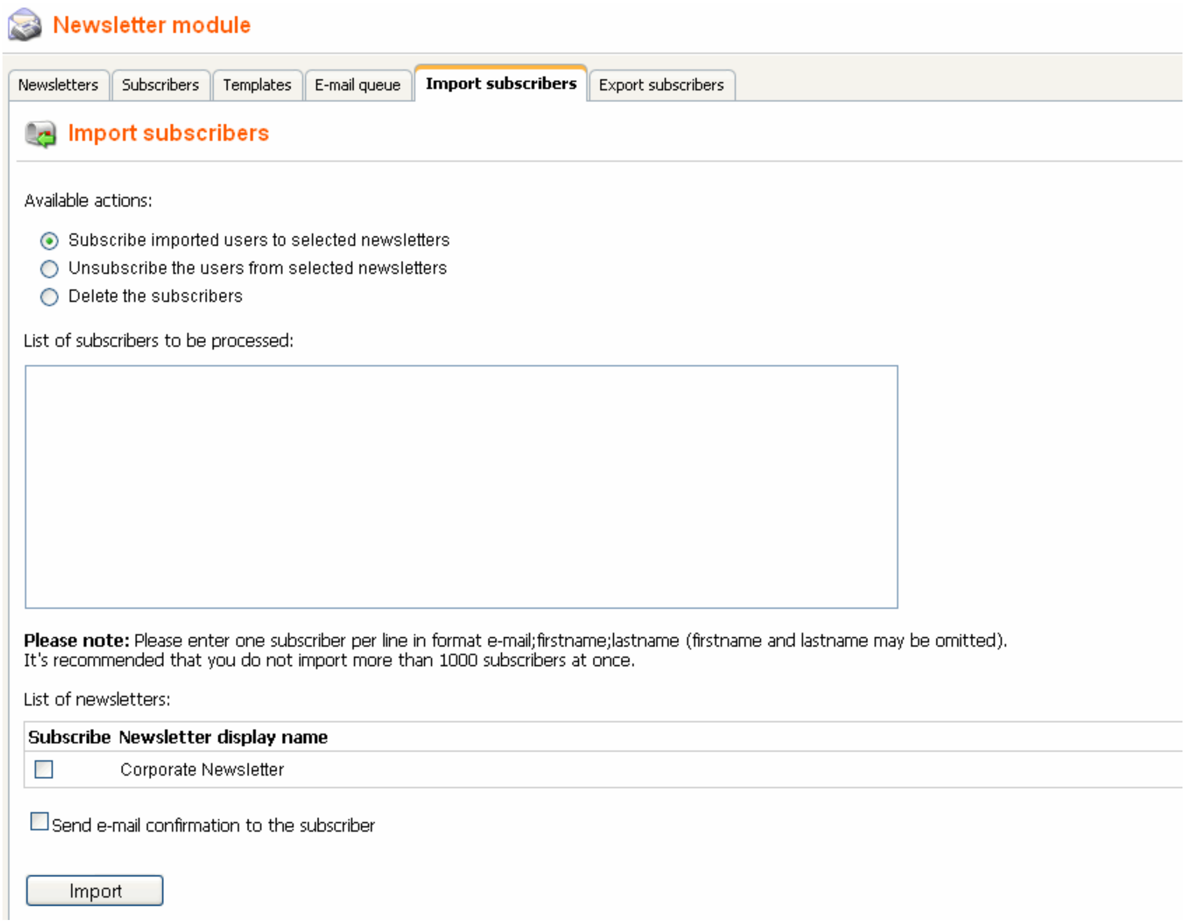
### Problems with role subscribers

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 user role. To set the newsletter for using the e-mail queue, go to **CMS Desk -> Tools -> Newsletter**, choose to **Edit** () your newsletter, switch to its **Configuration** tab and check the **Send issues via e-mail queue** check-box.

# 48.7 Subscriber import and export

## Importing subscribers

Import of subscribers can be carried out using the **CMS Desk -> Tools -> Newsletters -> Import subscribers** dialog.




You need to prepare a list of subscribers 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
marry.jones@mail.com;;Jones
frank.maguire@web.com;Frank
monica@italy.com
```

By selecting one of the tree radio buttons above the text area in combination with some of the newsletters in the list below it, the following actions can be done:

- **Subscribe imported users to the selected newsletters** - subscribes imported users to newsletters selected in the list below

- **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

Click **OK** to import the subscribers. If one of the lines contains an invalid entry, the import is not processes 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.

### 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.

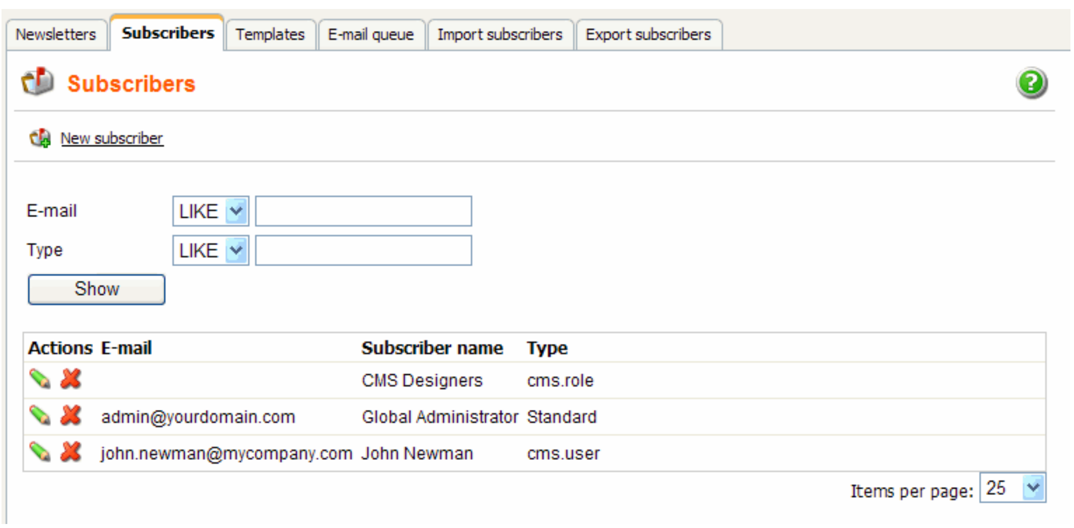
You can choose if you want to export all subscribers (do not specify any newsletter) or only subscribers of chosen newsletter(s). The subscribers are exported in format:

**email;firstname;lastname**

Then, you can copy and paste the list of subscribers from the textbox to your application.

## 48.8 Subscriber management

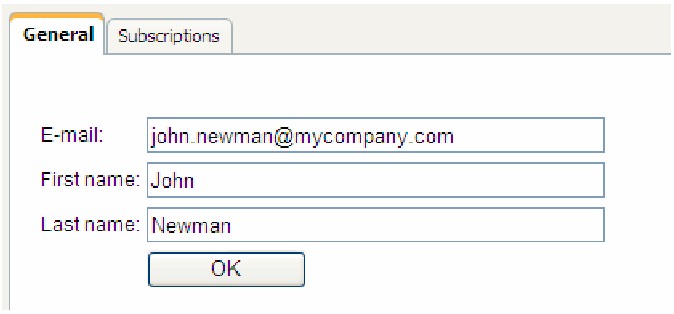
You can manage subscribers in **CMS Desk -> Tools -> Newsletter -> Subscribers**.



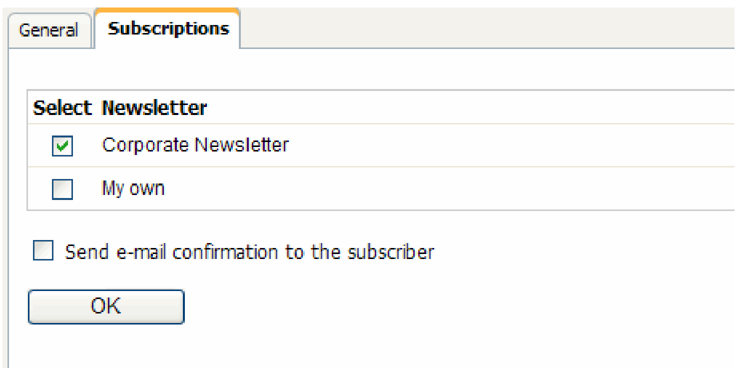

At the bottom of the page, you can see a list of all subscribers of any of the website's newsletters. You can filter these by e-mail or type, by searching for a text entered in the appropriate search text box and clicking the **Show** button.

By clicking the **Delete** icon next to a subscriber record, you can remove the subscriber from the list.

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.

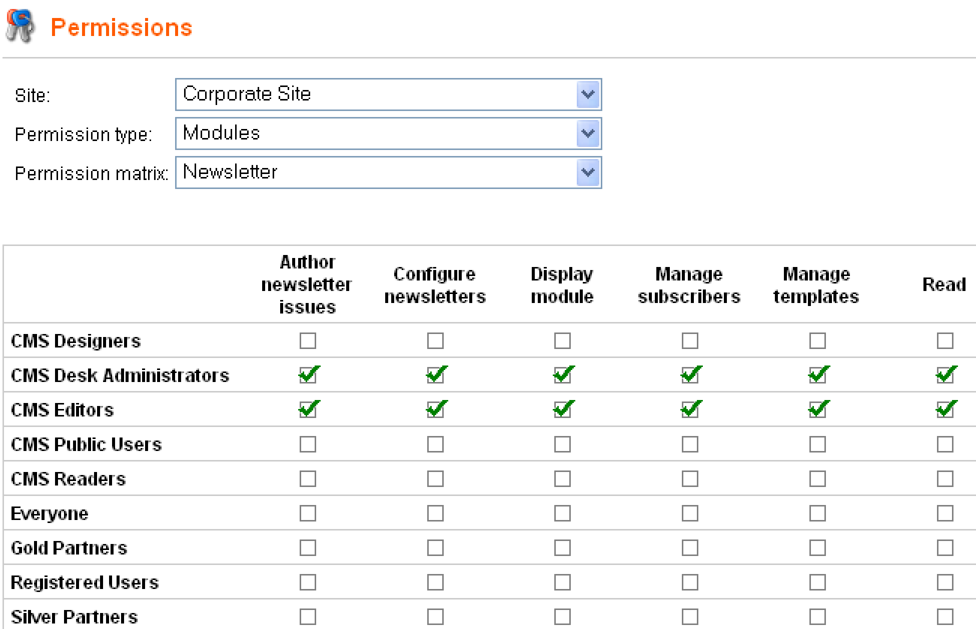



On the **Subscriptions** tab, you can select to which of all newsletters will this subscriber be subscribed. If you check the **Send e-mail confirmation to the subscriber** check-box, a notification e-mail will be sent to the user, informing him about the subscription changes.



## 48.9 Security

You can control the access to the Newsletter module in **CMS Desk -> Administration -> Permissions**, after you select **Modules -> Newsletter** permission matrix:




The following table explains particular permission names:

| | |
|---|---|
| Author newsletter issues | The user is allowed to create and edit newsletter issues. |
| Configure newsletters | The user is allowed to configure newsletter settings. |
| Display module | The module is visible in the user interface. |
| Manage subscribers | The user is allowed to create/edit/delete subscribers. |
| Manage templates | The user is allowed to create/edit/delete e-mail templates used by the Newsletter module. |
| Read | The user is allowed to read all data of the Newsletter module. |

# Part XX

# 49 Module Notifications

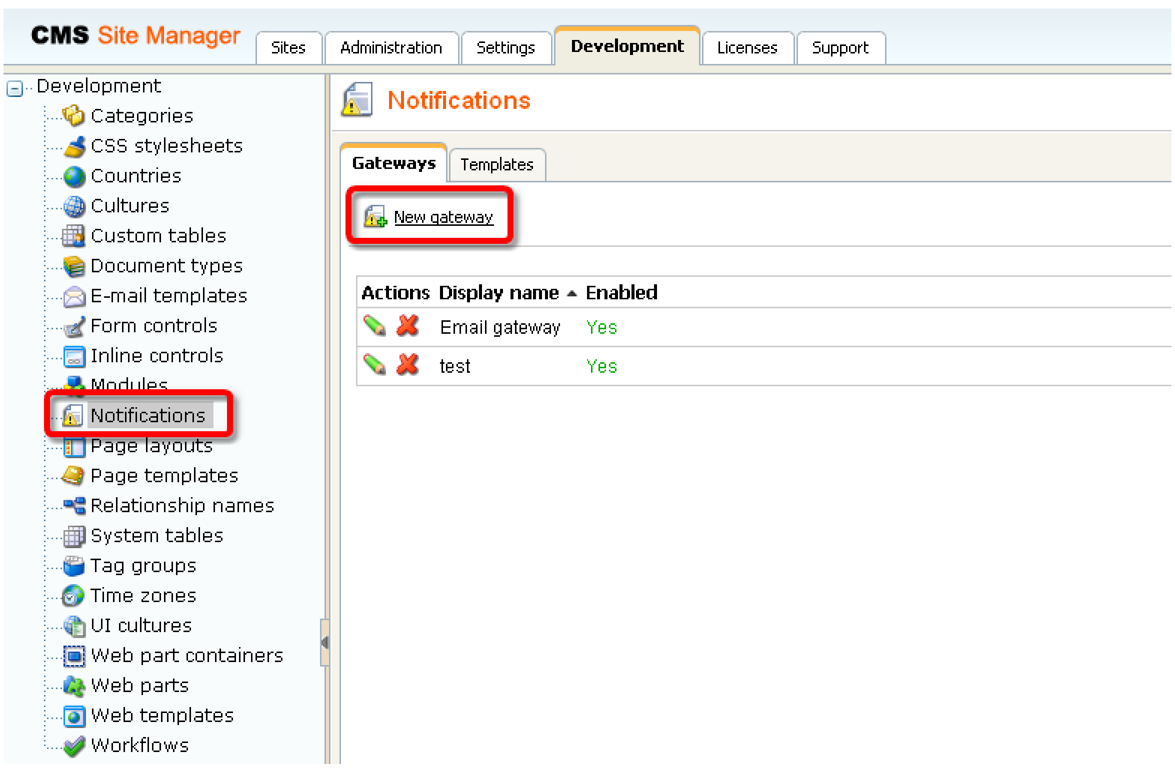
## 49.1 Overview

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 custom gateway, such as an SMS or ICQ provider. Site visitors or CMS Desk administrators can subscribe to receiving notifications about specific events and choose the providers they want to obtain notification message from. It means that for one event, user can obtain notification messages from multiple providers. By default, there are several built-in events (document updated, document deleted, etc.), however, you can create and use your custom ones.

## 49.2 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.

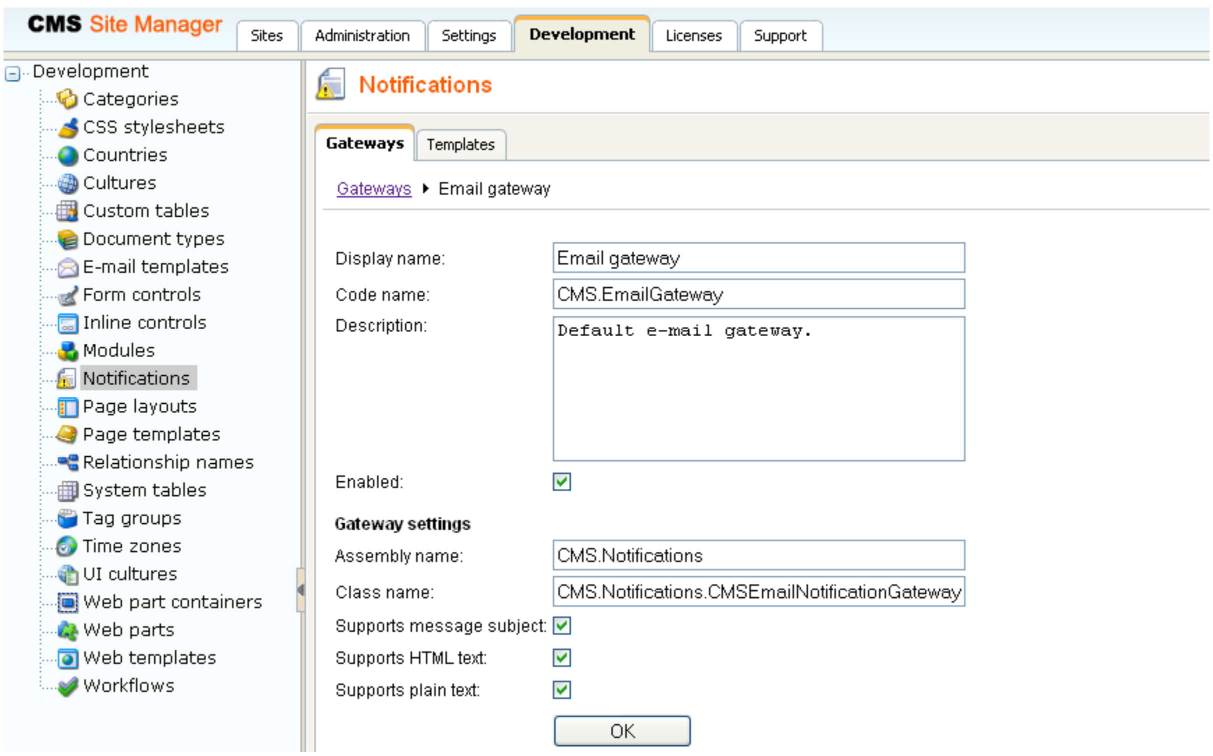
1) Go to **Site Manager -> Development -> Notifications**. On the **Gateways** tab, you can find the **New gateway** link at the top of the page. Click it, the New gateway form will be displayed.



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

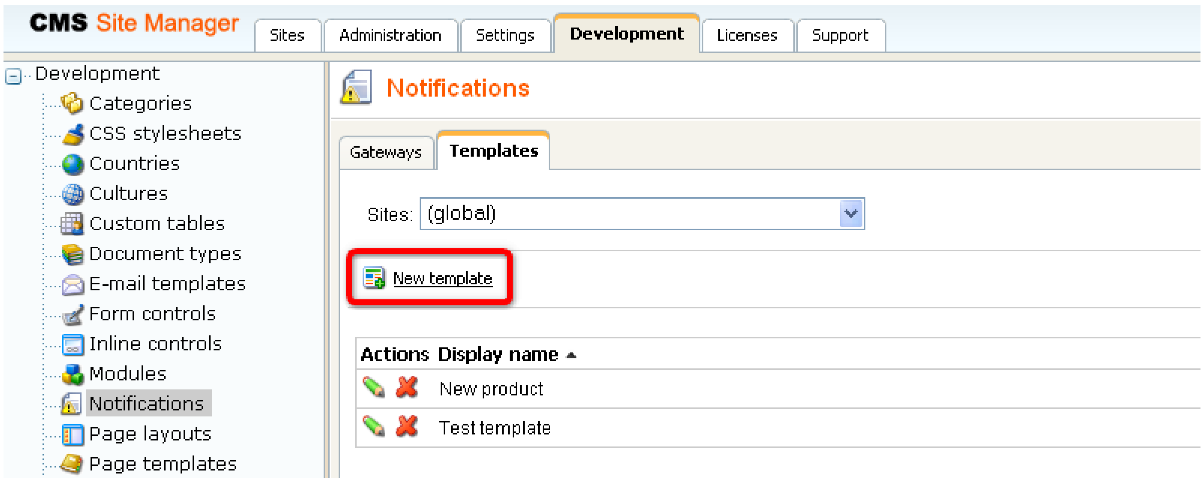
- **Assembly name** - name of the assembly in that the gateway code is stored
- **Class name** - name of the the class containing the gateway code; must be entered including the assembly name, as you can see in the screenshot below
- **Supports message subject** - check if the gateway's message format supports message subjects
- **Supports HTML text** - check if the gateway supports messages in HTML format (e.g. for e-mails)
- **Supports plain text** - check if the gateway supports plain text format (e.g. for SMS)




3) After entering all the details, click **OK**. You will be redirected back to the notification gateways list and you should be able to see your new gateway in the list.

## 49.3 Creating a notification message template

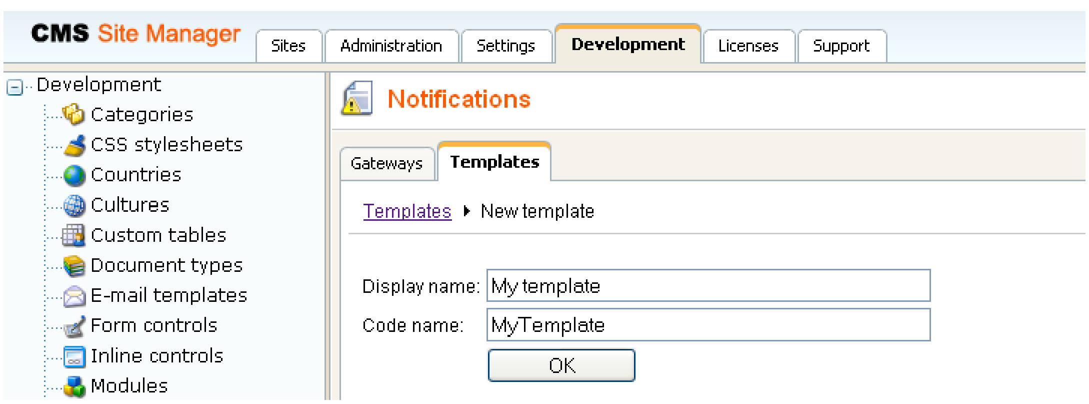
1) Go to **Site Manager -> Development -> Notifications** and switch to the **Templates** tab. Using the **Sites** drop-down list, you can select which site will be the template used by. If you choose **(global)**, the template will be available for all sites. When selected, click the **New template** 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.



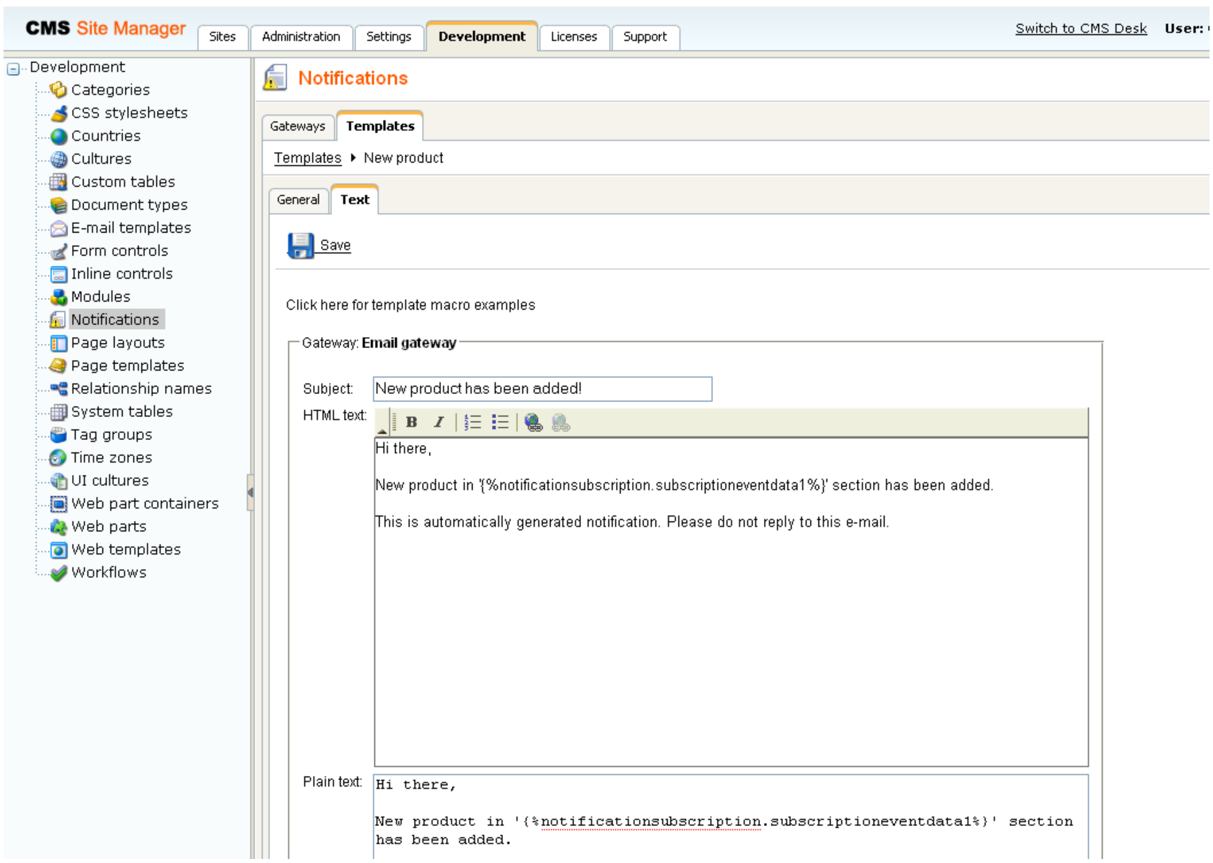

3) Now you should see the same form that you've just filled, but in a **General** tab, while another tab called **Text** is also available. Switch to the **Text** tab.

On this tab, 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 specified data column from the Notification_Gateway table; this represents notification gateway which performs sending notification message
- **{%notificationuser.UserID%}** - displays the value of specified user data column from the CMS_Usertable table; this represents user the notification message is sending to
- **{%notificationcustomdata.XXX%}** - displays the value of specified data column from 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)



## 49.4 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 sample Corporate site 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.

In order for you to see the full functionality, it is necessary to have 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 system 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** mode.

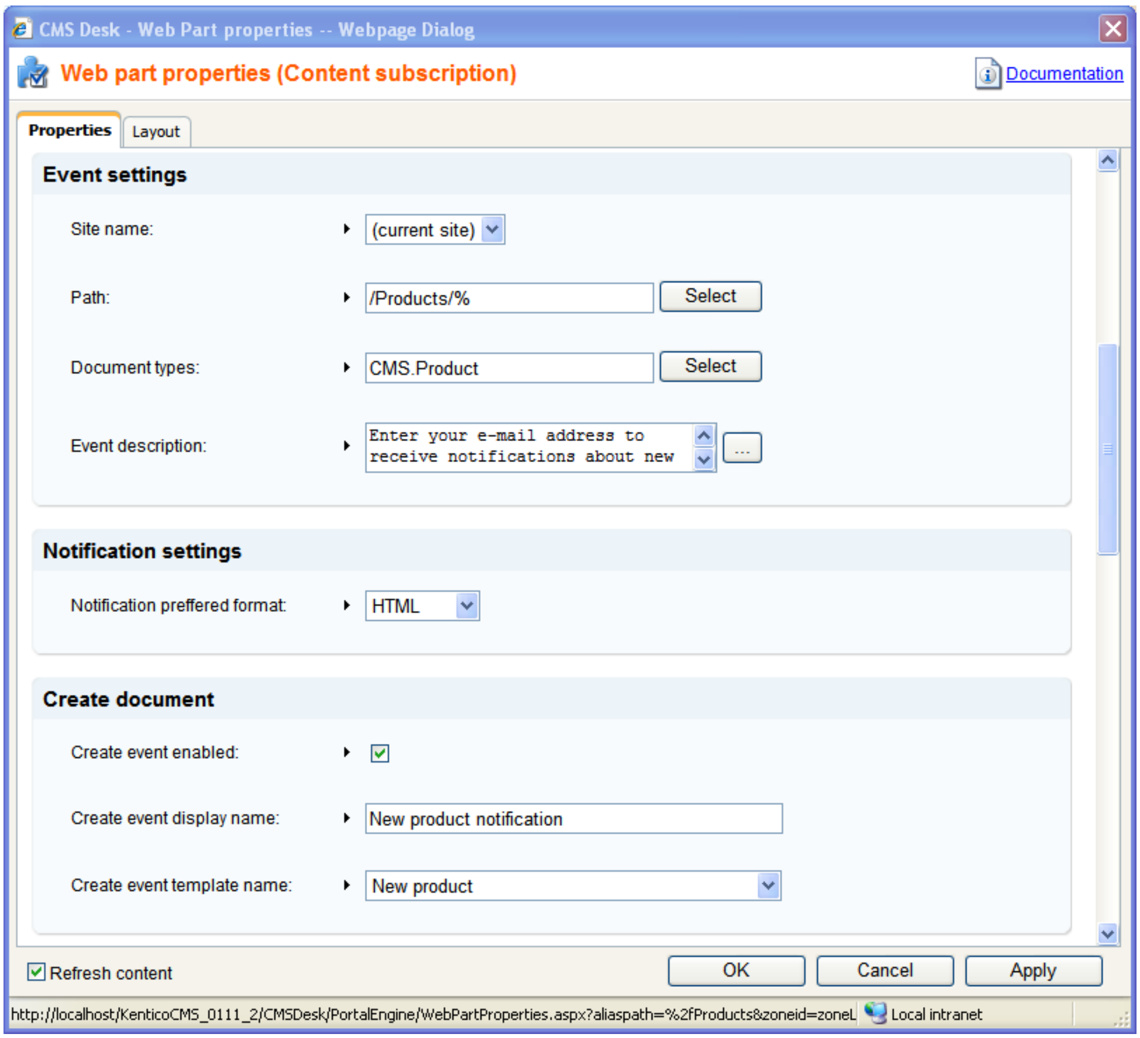
2) Add the **Content subscription** web part to the zoneLeft web part zone,  below the Left tree menu. In the Web part properties dialog, set the following properties:

- **Site name** - (current)
- **Path** - /Products/%
- **Document type** - CMS.Product
- **Event description** - Enter your e-mail address to receive notifications about new products:

- **Notification preferred format** - HTML

- **Create event enabled -** true
- **Create event display name** - New product notification
- **Create event template name** - New product

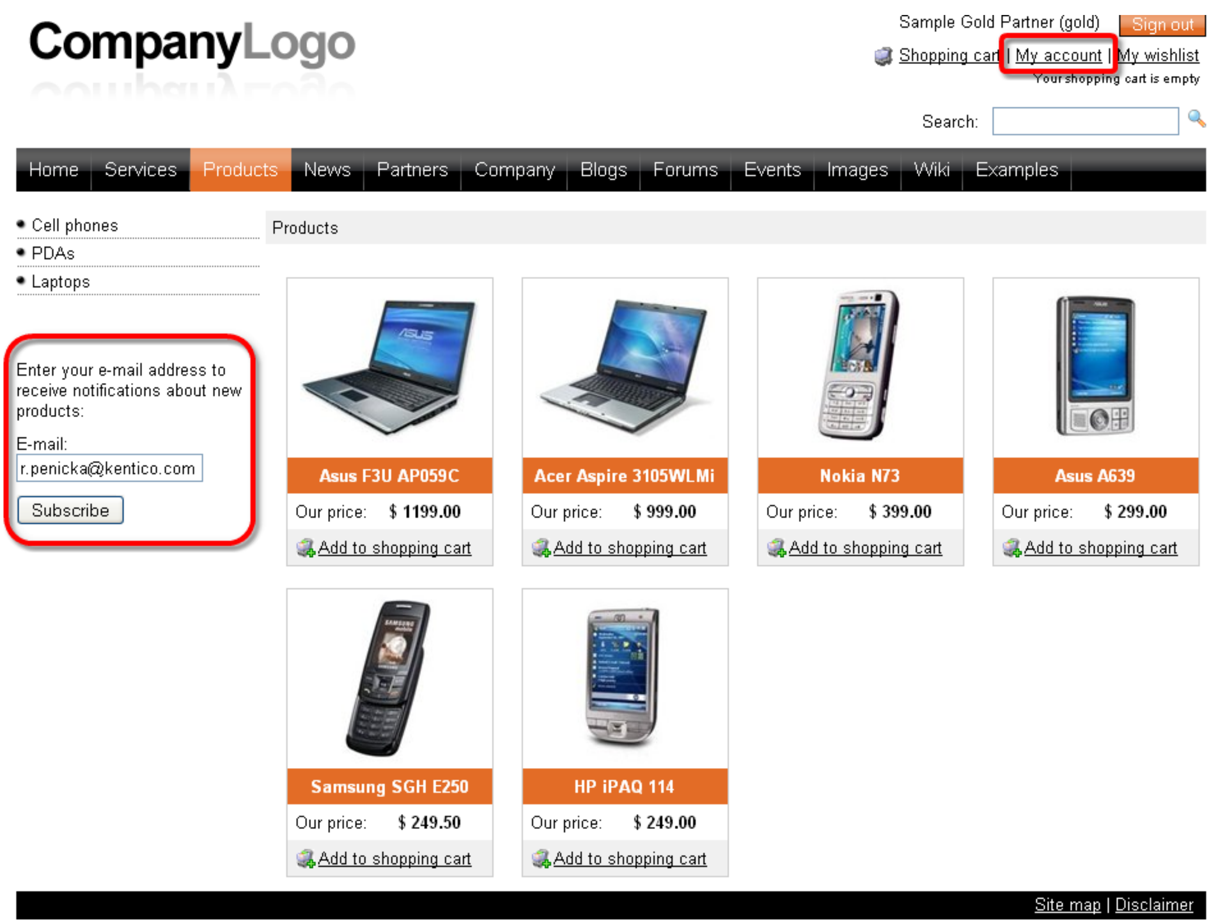
and click **OK**.



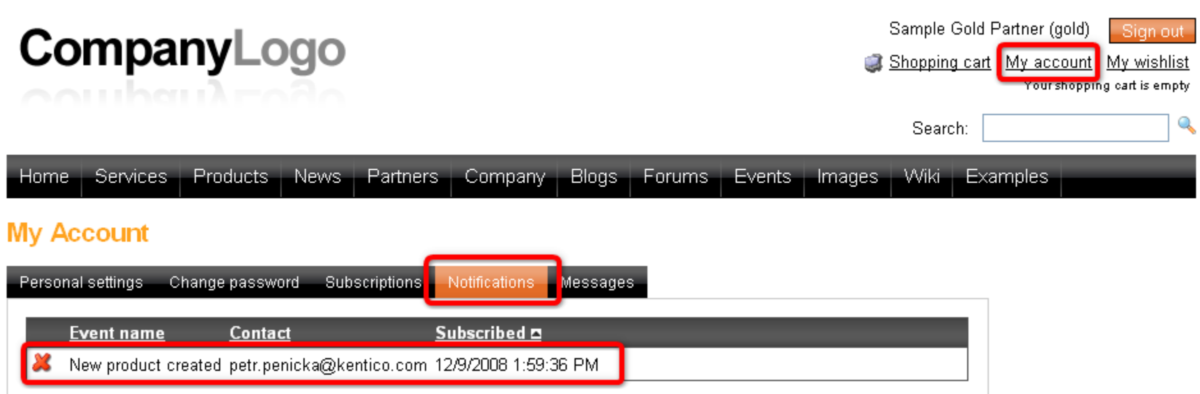

3) Let's switch to the **site visitor's** perspective now. Sign out of CMS Desk.

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.



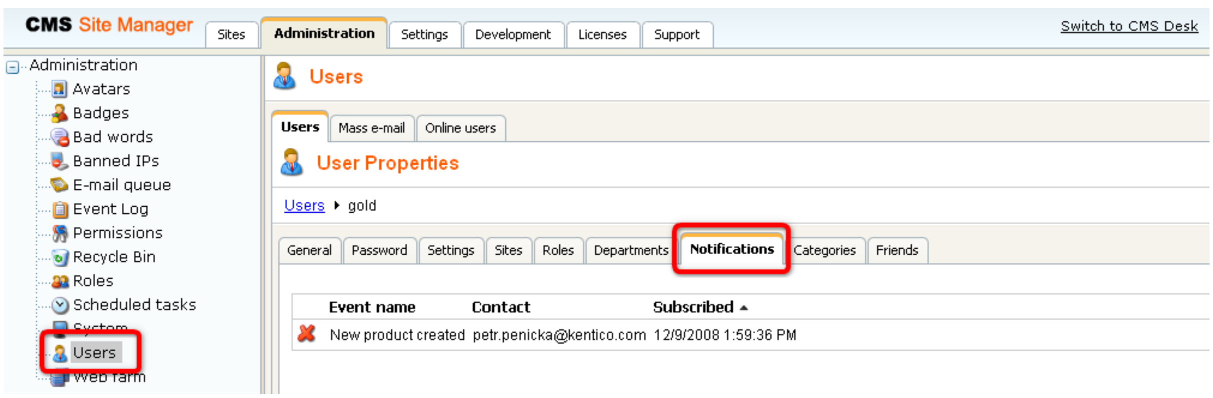

4) You can immediately verify your subscription by going to the **My account** section again. If you switch to the **Notifications** section of the My account web part, you should see the notification subscription present.



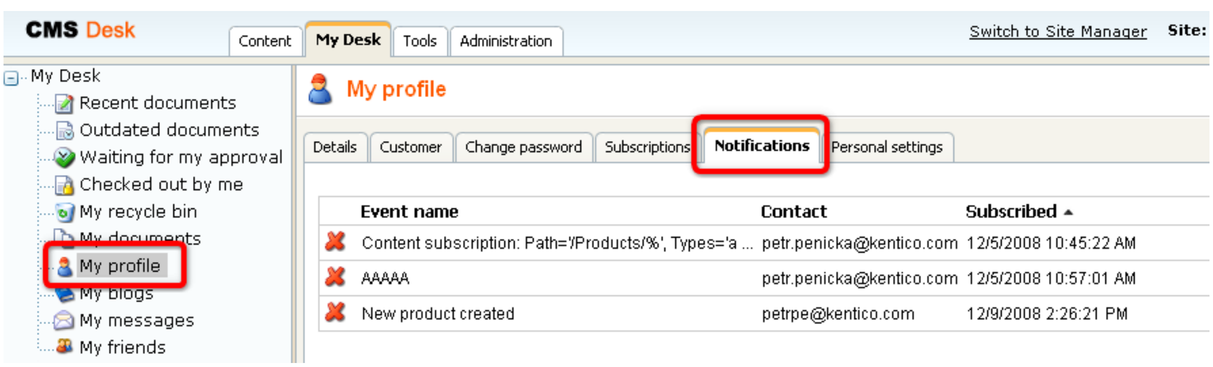
5) Now you can verify that the whole setup works. Log in to **CMS Desk** as the *administrator* again. From the content tree, select **Products** and click the **New** icon above the content tree. Choose to create some new product, e.g. a Cell phone, enter some sample data about the cell phone and click **Save**. Check your e-mail inbox in a few minutes. You should have received a new notification message about the newly added product.

## 49.5 Managing users' notifications

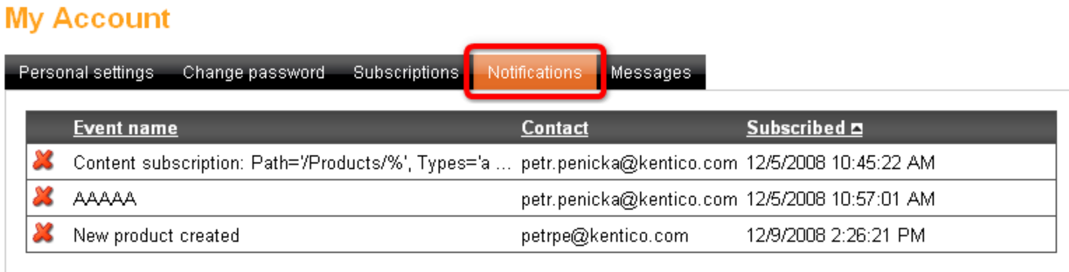
Site administrators can manage subscriptions of particular users in **Site Manager -> Administrator -> Users**. If you choose to **Edit** ( ) a user and switch to the **Notifications** tab, you will see all notifications that the current user is subscribed to. You can unsubscribe the user from a notification 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**.



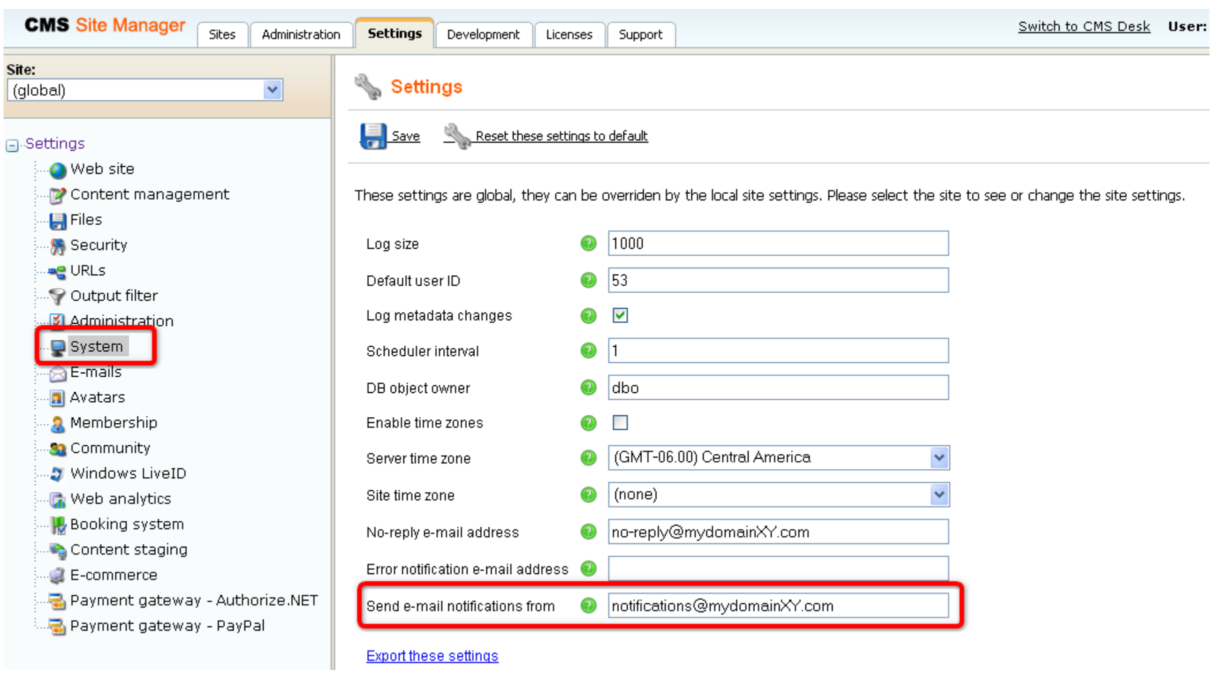
Site visitors registered to the site can manage their subscriptions the same way using the **My account** web part, on its **Notifications** tab.




## 49.6 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 ('From' field) of notification e-mails




## 49.7 Security

This 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.

# Part XXX

# 50 Module On-line users

## 50.1 Overview

The On-line users module allows monitoring of users connected to the web site. The module comes with the On-line users web part that can be used for displaying information about on-line users.

The module identifies a new user when a new session between the client browser is started. The user is considered off-line if the session expires or when the user logs off. This means that the user is still considered on-line when she closes her web browser without signing off.

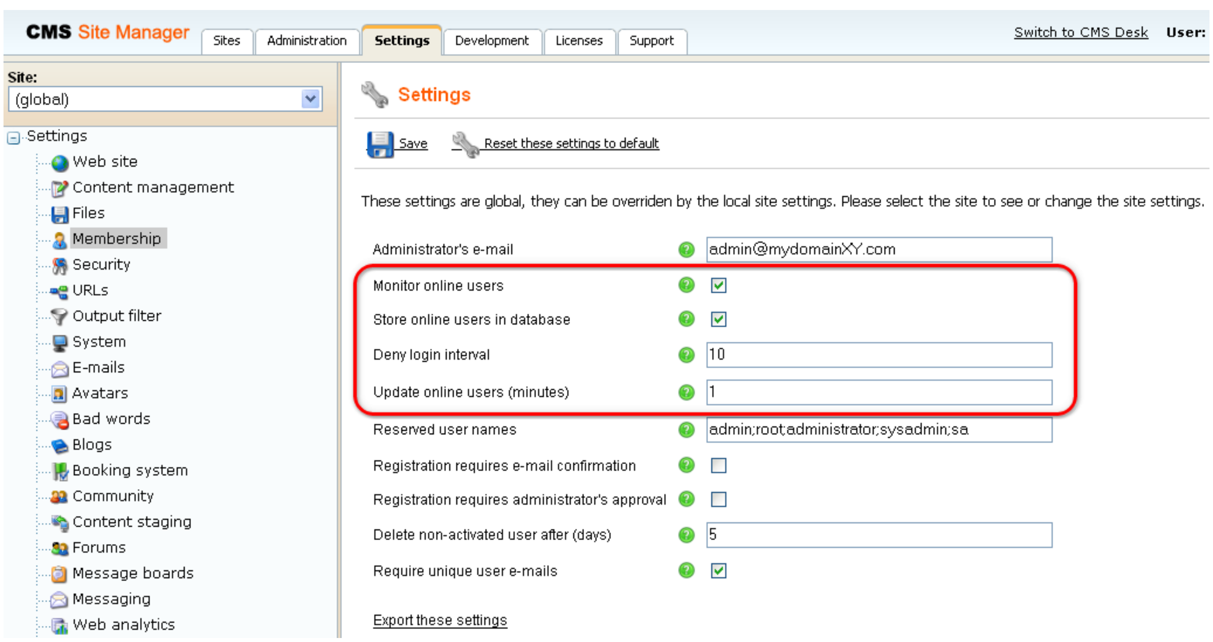
## 50.2 Enabling the module

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 that information about on-line users will be saved in the database.

The **Deny login interval** property determines how long will users not be able to log-in when they are kicked. The value is entered in minutes.

Finally, the **Update online users (minutes)** property defines how often will information about users accessing the site be 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 **Sessions remove expired sessions** scheduled task (you can read the value in Site Manager -> Administration -> Scheduled tasks -> edit ( ) Sessions remove expired sessions -> Task interval -> Every: X minutes).



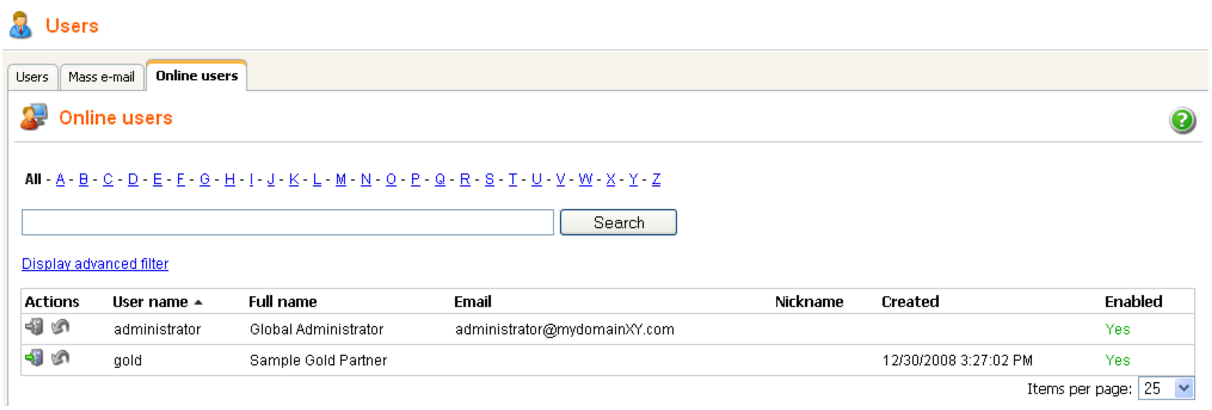
## 50.3 On-line users tab

If you go to **Site Manager -> Administration -> Users** and switch to the **On-line users** tab, you can see a list of all users that are currently accessing the web site. 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 specified criteria. The filter can work in two modes between which you can switch by clicking the **Display advanced filter**, respectively **Display simplified search** links.

### Simplified search

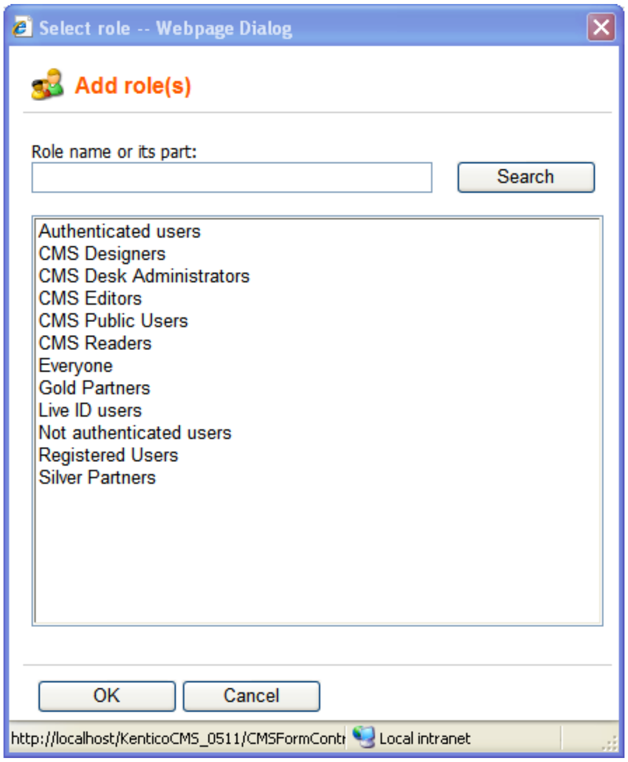
This option offers only one field. The expression entered into the field will be searched in users' User Name, Full Name, E-mail Address and Nickname properties.



### Advanced filter

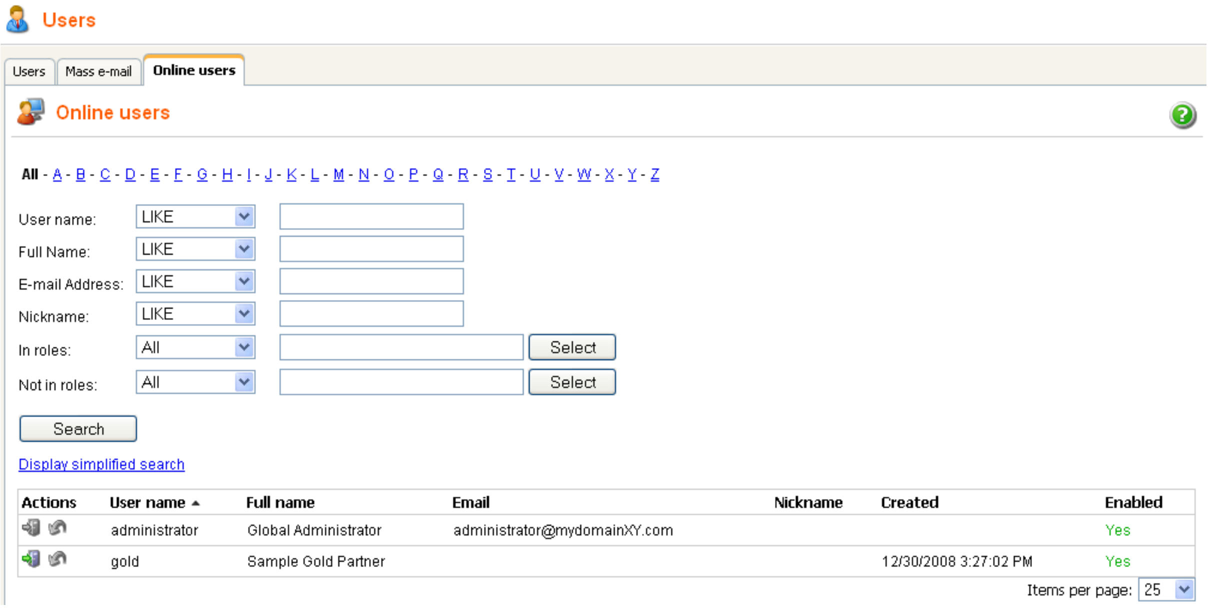
The advanced filter offers searching by User Name, Full Name, E-mail Address and Nickname. You can also filter on-line users by their belonging to roles. You can specify which roles are (**In roles**) and are not (**Not in roles**) the listed users members of.

You can type in code names of the user roles manually into the fields, divided by a semicolon (;). Using the **Select** buttons, you can open the role selection pop-up window. SHIFT and CTRL can be used for multiple selection. Click **OK** to confirm the selection.



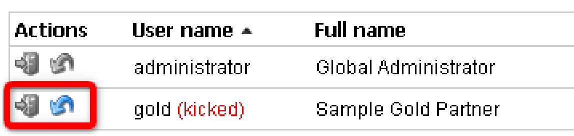

When you have entered all search criteria, click the **Search** button.

Only those users that match the specified criteria will be listed.



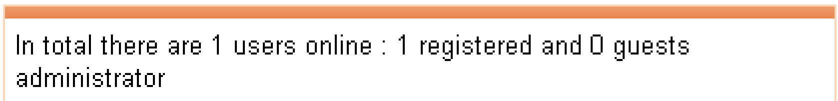

## Kicking a user

If you click the **Kick** ( ) icon next to some of the listed users, you can kick the user out of the web site, which means that the user will be logged out of the web site. 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 it easily by clicking the **Take back** ( ) icon.



## 50.4 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** folder. 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 the list of on-line users that is displayed based on the transformation defined by the **Transformation name** property.

The web part has the following specific properties:

| | |
|---|---|
| Transformation name | Name of the transformation that will be used to display on-line users.<br><br>If you choose to Select the transformation, you can find the default transformation that has been used in the screenshot above under Document type -> CMS.Root -> 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.<br><br>You can use the following formatting macros that will be substituted by the appropriate number:<br><br>{0} - number of all connected users<br>{2} - number of connected registered users<br>{1} - number of connected anonymous users |
| No users on-line text | Text that will be displayed in case that no users are currently online. |

# 50.5 API

Session management methods are located in the **SessionManager** class found in the **CMS. CMSHelper** library.

For detailed API documentation, please refer to **Kentico CMS API Reference** guide that is a part of your Kentico CMS installation and can be accessed through the program folder in Windows Start menu.

Let's mention the following three methods that are the most important ones:

| | |
|---|---|
| GetOnlineUsers(String, String, Int32, String, String) | Returns UserInfos of online users according to the WHERE condition and ordered by the ORDER BY expression.<br><br>String - WHERE condition<br>String - ORDER BY expression<br>Int32 - TOP N expression<br>String - location (alias path) of users on the web site<br>String - site name |
| IsUserOnline(Int32) | Returns true when the user with the specified ID is online.<br><br>Int32 - ID of the user |
| KickUser(Int32) | Performs all the necessary actions to kick the user with the specified ID.<br><br>Int32 - ID of the user |

# Part XX

# 51 Module Polls

## 51.1 Overview

The Polls module allows you to create and publish polls with single or multiple answers.


The polls can be managed in the **CMS Desk -> Tools -> Polls** dialog.

The polls module uses a cookie to ensure that the same visitor (using the same browser) cannot vote twice in the same poll.

## 51.2 Managing Polls

You can manage polls in **CMS Desk -> Tools -> Polls**.

### Creating a new poll

Click **New poll** and fill in 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. |

Click **OK**.

### Editing poll properties

Now you can 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. |

### Defining answers

Then, you can define the list of available answers on the **Answers** tab. To create a new answer, click **New answer** and enter the answer text.

You can also choose if the given answer should be enabled which means it's displayed in the list of options - this is useful if you need to remove some answer from the poll while keeping the number of votes in the history. The disabled option is then not calculated into the displayed results.

### 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 web sites so that they can put the poll into the text through 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 web site may use different design depending on the design of your web site and may behave differently depending on the web part settings (if you publish it using a web part).

## 51.3 Publishing Polls

There are two ways how you can publish a poll on the web site:

- **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

Once you define 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 **Insert Poll** 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're 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 choose the Poll name and then you can configure some additional settings of the poll that are described in **Kentico CMS Web Parts and Controls Reference**.

If you're 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 web site. You can do that in **CMS Desk -> Tools -> Polls**. Click the **Sites** tab and assign the poll to the required site.

## 51.4 Multilingual support

When you use the Polls module on a multilingual web site, 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 web site.

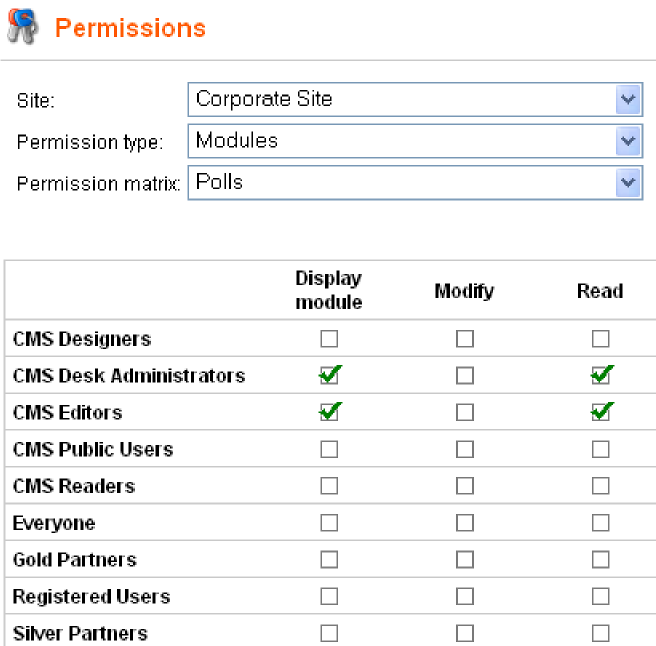
  or

- You can create **a single poll for all languages** - in this case, all votes are tracked together. In order to localize the texts of the poll, you need to use Localization Expressions.

# 51.5 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.




| | Display module | Modify | Read |
|---|---|---|---|
| CMS Designers | ☐ | ☐ | ☐ |
| CMS Desk Administrators | ☑ | ☐ | ☑ |
| CMS Editors | ☑ | ☐ | ☑ |
| CMS Public Users | ☐ | ☐ | ☐ |
| CMS Readers | ☐ | ☐ | ☐ |
| Everyone | ☐ | ☐ | ☐ |
| Gold Partners | ☐ | ☐ | ☐ |
| Registered Users | ☐ | ☐ | ☐ |
| Silver Partners | ☐ | ☐ | ☐ |

You can choose which user roles are authorized to **read** poll details and **modify** polls (includes create/modify/delete). The **Display module** permission makes the module visible in the user interface.

## 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:

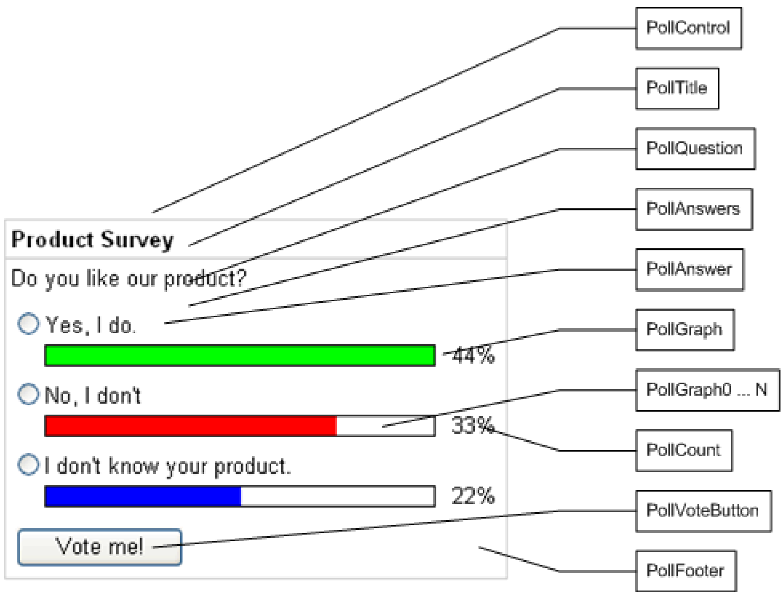
- **All users** - any visitor can vote
- **Only authenticated users -** only site members who sign in can vote
- **Only authorized roles** - only authenticated members of chosen roles can vote

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.

## 51.6 Design and styles

The design is controlled by CSS styles included in your CSS stylesheet for the given site. The following figure shows the polls styles:



# 51.7 Polls API and Internals

## 51.7.1 Database tables and API classes

The Polls module uses the following 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 - manage answers
- PollInfo, PollInfoProvider - manage polls

The following chapters show how to use these classes.

## 51.7.2 Creating a new poll

The following sample code shows how you can create a new poll using the API:

**[C#]**

```
using CMS.Polls;
using CMS.SettingsProvider;
using CMS.CMSHelper;
using CMS.DataEngine;
using CMS.GlobalHelper;

...

        // 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 = PollInfoProvider.ACCESS_ALL;
        pollObj.PollOpenFrom = DateTime.Now;
        pollObj.PollOpenTo = DateTime.Now.AddMonths(1);

        // Save the object
        PollInfoProvider.SetPollInfo(pollObj);

        // Add poll to current site
        if (CMSContext.CurrentSite != null)
        {
            PollInfoProvider.AddPollToSite(pollObj.PollID, CMSContext.CurrentSite.SiteID);
        }
```

## 51.7.3 Updating an existing poll

The following sample code shows how you can modify poll properties using the API:

**[C#]**

```
using CMS.Polls;
using CMS.SettingsProvider;
using CMS.DataEngine;
using CMS.GlobalHelper;

...

        string pollName = "testingpoll"; // Testing poll code name

        // Get poll info by code name
        PollInfo pollObj = PollInfoProvider.GetPollInfo(pollName);

        if (pollObj != null)
        {
            // Set the field values
            pollObj.PollResponseMessage = "Thank you for your time.";
            pollObj.PollAllowMultipleAnswers = true;

            // Save the changes
            PollInfoProvider.SetPollInfo(pollObj);
        }
```

## 51.7.4 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;
using CMS.SettingsProvider;
using CMS.DataEngine;
using CMS.GlobalHelper;

...

        string pollName = "testingpoll"; // 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(pollObj.PollID) + 1;
            pollAnswerObj.AnswerCount = 0;
            pollAnswerObj.AnswerPollID = pollObj.PollID;
            pollAnswerObj.AnswerEnabled = true;
            pollAnswerObj.AnswerText = "Yes, I love it.";

            // Save the answer
            PollAnswerInfoProvider.SetPollAnswerInfo(pollAnswerObj);
        }
```

### 51.7.5 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;
using CMS.SettingsProvider;
using CMS.DataEngine;
using CMS.GlobalHelper;

...

        string pollName = "testingpoll"; // 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[0]);

                if (pollAnswerObj != null)
                {
                    // Increment votes by 1
                    pollAnswerObj.AnswerCount++;

                    // Save the data
                    PollAnswerInfoProvider.SetPollAnswerInfo(pollAnswerObj);
                }
            }
        }
```

### 51.7.6 Add authorized roles

The following sample code shows how you can add roles authorized for voting.

**[C#]**

```
using CMS.Polls;
using CMS.SettingsProvider;
using CMS.SiteProvider;
using CMS.DataEngine;
using CMS.GlobalHelper;

...

        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);
            }
        }
```

### 51.7.7 Deleting a Poll

The following sample code shows how you can delete a poll including its answers.

**[C#]**

```
using CMS.Polls;

...

        string pollName = "testingpoll";

        // Get poll info for the specified code name
        PollInfo pollObj = PollInfoProvider.GetPollInfo(pollName);

        if (pollObj != null)
        {
            // Delete the poll
            PollInfoProvider.DeletePollInfo(pollObj.PollID);
        }
```

## 51.7.8 Deleting a Poll Answer

The following sample code shows how you can delete a poll answer.

**[C#]**

```
using CMS.Polls;
using CMS.SettingsProvider;
using CMS.DataEngine;
using CMS.GlobalHelper;

...

        string pollName = "testingpoll";

        // 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].Rows[0]);

                if (pollAnswerObj != null)
                {
                    // Delete poll answer
                    PollAnswerInfoProvider.DeletePollAnswerInfo(pollAnswerObj.AnswerID);

                    lblInfo.Text = "The poll answer was deleted successfully.";
                }
                else
                {
                    this.lblInfo.Text = "Failed to delete the poll answer.";
                }
            }
            else
            {
                this.lblInfo.Text = "No poll answers found.";
            }
        }
```

# Part XX

# 52 Module Reporting

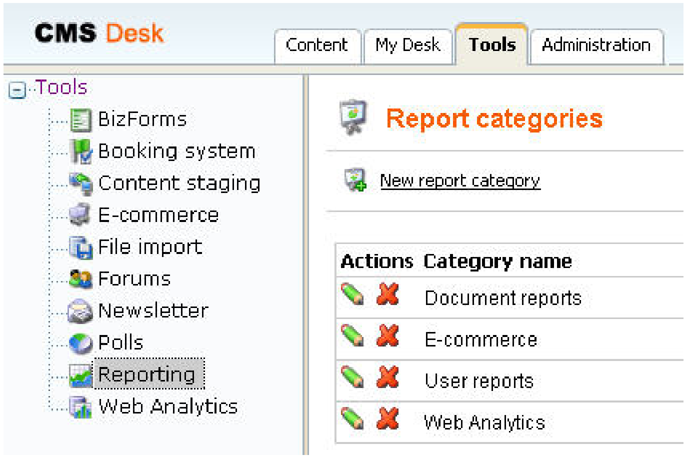
## 52.1 Overview

The Reporting module allows you to create internal reports to watch the activity in the Kentico CMS system and on the web site, such as recently created documents, expired documents, web site visits, etc.. It's also used in the Site Analytics module.

You can manage the reports in **CMS Desk -> Tools -> Reporting**.

## 52.2 Managing report categories

All reports are organized in categories. You can manage the categories in the Reporting module.




**Example:**

Click New report category and enter the following values:

- **Category display name:** User reports
- **Category code name:** UserReports

Click **OK**.

## 52.3 Creating a new report

Click edit next to some of the report categories and choose the **Reports** tab. 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.

### Editing report layout

Enter the following text in the layout editor: **Pages by page template**

Select the text and set its format to **Heading 1**. Click **OK** to save changes.

## Adding a table

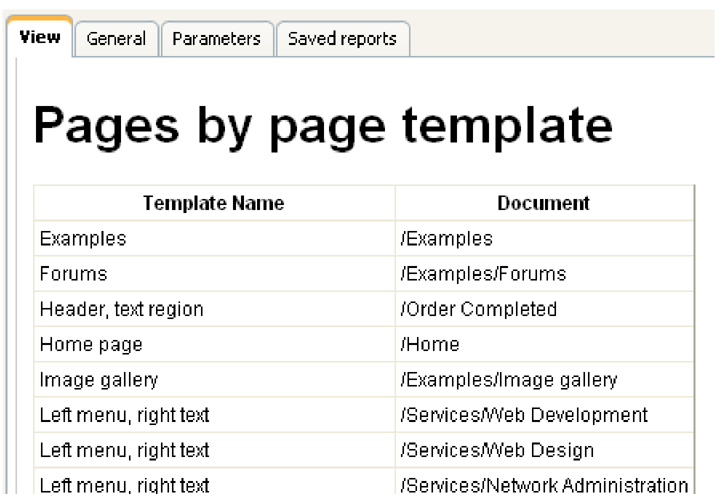
Click **Add** in the **Tables** section on the right. Enter the following values:

- **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

Click **OK**. Place the cursor in the layout editor under the title, choose the table in the **Tables** section and click **Insert**. Click **OK** to save changes and switch to the **View** tab. You will see a report like this:




| Template Name | Document |
|---|---|
| Examples | /Examples |
| Forums | /Examples/Forums |
| Header, text region | /Order Completed |
| Home page | /Home |
| Image gallery | /Examples/Image gallery |
| Left menu, right text | /Services/Web Development |
| Left menu, right text | /Services/Web Design |
| Left menu, right text | /Services/Network Administration |

**Writing queries for tables**

The queries you write for the 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 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 column names from the returned data set. If you need to use user friendly names, you can use the following syntax:

```
SELECT PageTemplateDisplayName as [Template Name], ...
```

## Adding a Graph

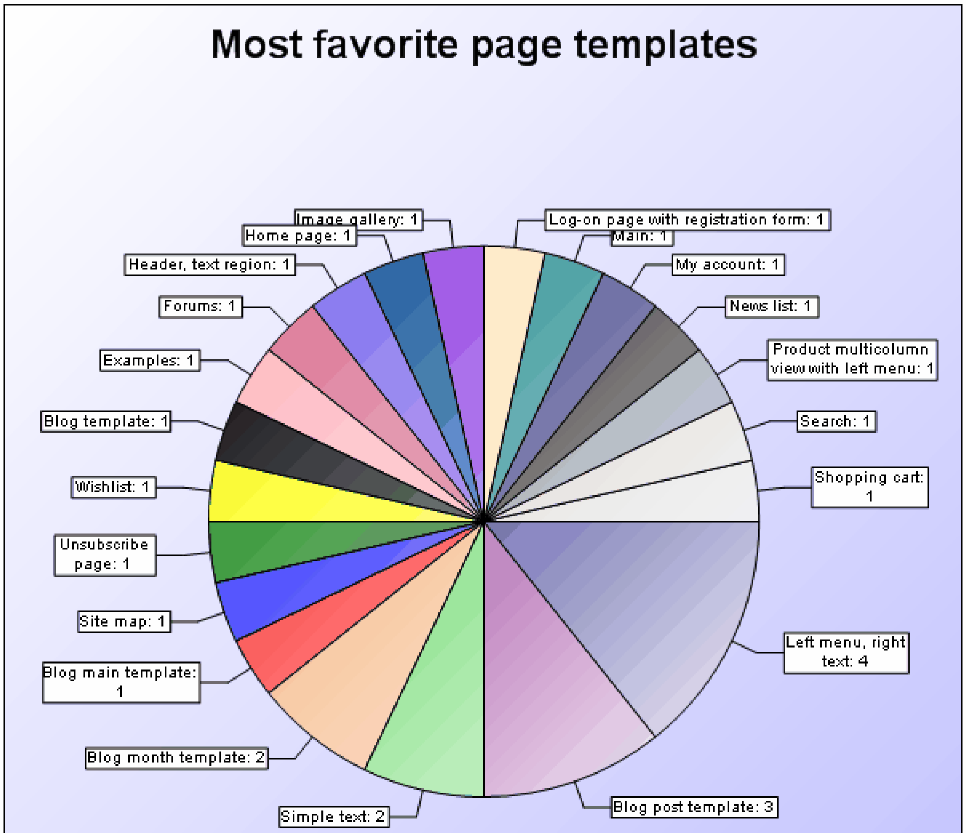
Switch back to the **General** tab. Click **Add** in the **Graphs** section on the right. Enter the following values:

- **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**. Place the cursor in the layout editor under the table, select the graph in the **Graphs** section and click **Insert**. Click **OK** to save changes and switch to the **View** tab. You will see a graph like this in the report:



Here is a complete list of graph properties and their descriptions:

| | |
|---|---|
| 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:<br>**Pie -** pie graph, accepts only one column for values, value in the third column is used for setting 'pie pieces' offset<br><br>**Bar -** bar graph, accepts multiple values and displays them next to each other<br><br>**Bar overlay** - bar graph, accepts multiple values and displays them behind each other with the lower values in the front<br><br>**Bar stacked** - bar graph, accepts multiple values and displays them on top of each other<br><br>**Bar percentage -** bar graph, accepts multiple values, displays one column for each category and divides it by percentage determined by the values<br><br>**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 divide ratio is displayed with the y-axis title |
| Show grid | Shows thin dotted line grid in the graph chart; 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 lines of the line graph |
| Title | Title of the graph |
| X axis title | Title of the horizontal axis in the graph |
| Y axis title | Title of the vertical axis in 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 font properties of the graph title |
| Axis font | Determines font properties of axis titles |

| | |
|---|---|
| X-axis angle | Determines declination angle of X-axis descriptions; setting this parameter to 90 causes upright descriptions |
| Y-axis angle | Determines declination angle of Y-axis descriptions |
| Scale min | Determines the minimum value from that will values in the graph be displayed |
| Scale max | Determines the maximum value up to that will values in the graph be displayed |
| Use colors | Determines colors used for graph items; accepts standard HTML color names or hexadecimal color codes divided by a semicolon; if lesser number of colors than is entered, remaining colors are generated automatically |
| Use symbols | Determines symbols used for values in the 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; The **From** and **To** fields accept standard HTML color names or hexadecimal color codes, the **Angle** field sets gradient declination |
| Chart gradient | 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 | 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 |

**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 pie pieces offset.

**Writing queries for bar charts**

The queries for the pie chart 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).

## Adding a Scalar Value

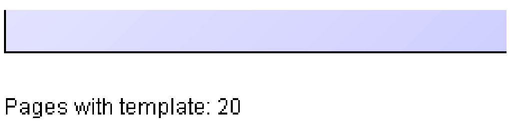
Click **Add** in the **Values** section on the right. Enter the following values:

- **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**. Place the cursor in the layout editor under the graph, choose the value in the **Values** section and click **Insert**. Click **OK** to save changes and switch to the **View** tab. You will see a text like this:




**Writing queries for scalar values**

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.

**Formatting scalar values**

You can format the displayed value using the standard .NET expressions. Examples:
- {0} - displays the value
- {0:F1} - displays number in format 10.5

You can find more details in .NET Framework documentation.

**Localizing strings in reports**

If you need to create a single report in multiple languages, please use the Localization Expressions.

## 52.4 Defining report parameters

The reports may be filtered using custom parameters. You can define the parameters on the **Parameters** tab in the **Report properties** dialog.

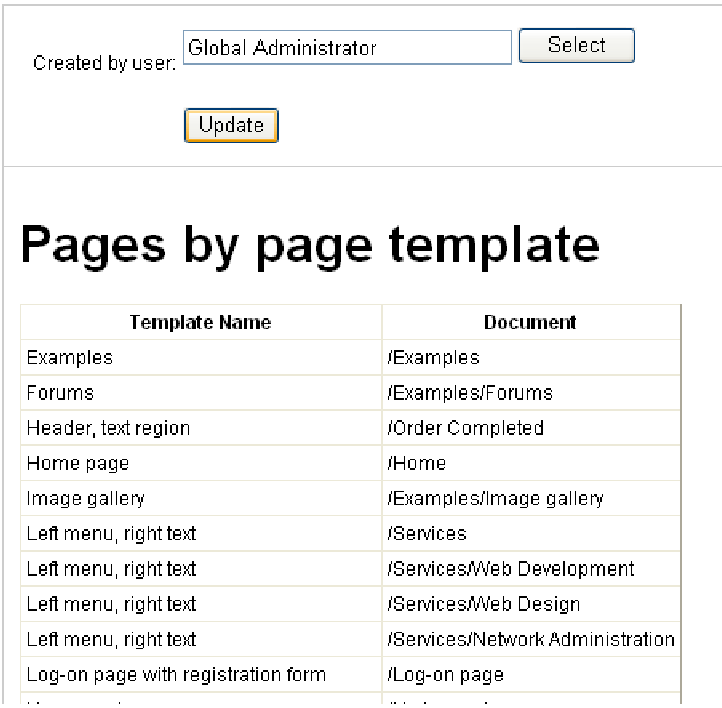
Click **New attribute** and enter the following values:

- **Attribute name:** UserID
- **Attribute type:** Integer number
- **Attribute default value:** 53
- **Field caption:** Created by user
- **Field type:** User Selector

Click **OK**. 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 and edit 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 the report with filter like this:



Created by user: Global Administrator Select

Update

Pages by page template

| Template Name | Document |
|---|---|
| Examples | /Examples |
| Forums | /Examples/Forums |
| Header, text region | /Order Completed |
| Home page | /Home |
| Image gallery | /Examples/Image gallery |
| Left menu, right text | /Services |
| Left menu, right text | /Services/Web Development |
| Left menu, right text | /Services/Web Design |
| Left menu, right text | /Services/Network Administration |
| Log-on page with registration form | /Log-on page |

### 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 the 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 web site, you can use 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 document expired on or before {%CMSContextCurrentTime%}:**

displays:

**List of document expired on or before 8/12/2007 12:06:49 PM:**

You can use this syntax for both custom report parameters and context paramters.

## 52.5 Saving report

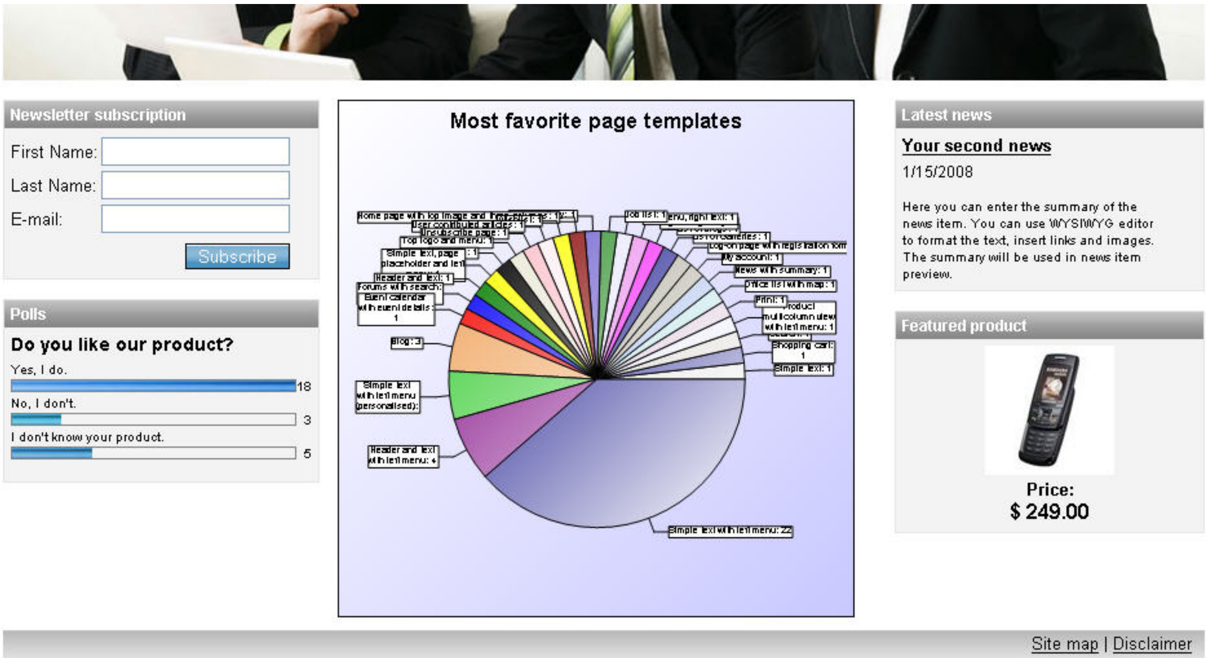
When you view some report, you can save it into the report history using the **Save** button. You can view the saved reports on the **Saved reports** tab. When the data change, the saved reports are not affected.

## 52.6 Displaying report on a web site

If you want to display some report on the web site 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"

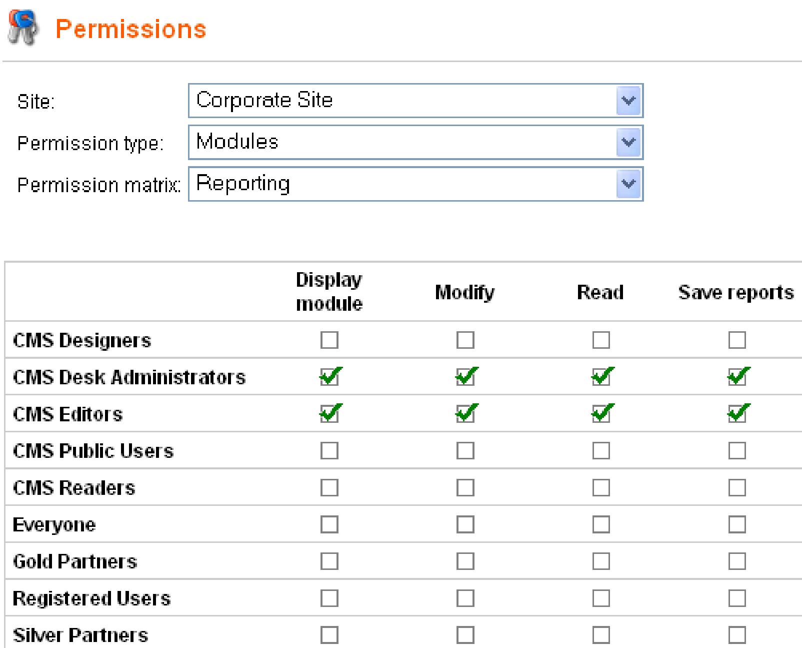
- **Report name** - select the required report
- **Display filter** - indicates if filter should be displayed on the page (if the report has some parameters specified)

The following figure shows a report displayed on the web site:



# 52.7 Security

You can configure the access to the Reporting module in **CMS Desk -> Administration -> Permissions**. Choose the permission matrix for **Modules** -> **Reporting.**




| | Display module | Modify | Read | Save reports |
|---|---|---|---|---|
| CMS Designers | ☐ | ☐ | ☐ | ☐ |
| CMS Desk Administrators | ✓ | ✓ | ✓ | ✓ |
| CMS Editors | ✓ | ✓ | ✓ | ✓ |
| CMS Public Users | ☐ | ☐ | ☐ | ☐ |
| CMS Readers | ☐ | ☐ | ☐ | ☐ |
| Everyone | ☐ | ☐ | ☐ | ☐ |
| Gold Partners | ☐ | ☐ | ☐ | ☐ |
| Registered Users | ☐ | ☐ | ☐ | ☐ |
| Silver Partners | ☐ | ☐ | ☐ | ☐ |

You can assign user roles roles with the following permissions:

- **Display module -** displays the module in the user interface
- **Modify** - allows users to create, modify and delete reports
- **Read** - allows users to read prepared reports
- **Save reports** - allows users to save reports

### Making reports available on the live site

The **Allow public users to see this report** property of the 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.

# 52.8 Reporting internals and API

## 52.8.1 Database tables and API classes

The Reporting module uses the following database tables:

- Reporting_Report - contains report settings and layout
- Reporting_ReportCategory - contains report categories
- Reporting_ReportGraph - contains graph definitions
- Reporting_ReportTable - contains table definitions
- Reporting_ReportValue - contains report definitions
- Reporting_SavedGraph - contains graphs saved with saved reports (in binary format)
- Reporting_SavedReport - contains saved reports

You can manage the reports using the classes in the CMS.Reporting namespace:

- ReportCategoryInfo, ReportCategoryInfoProvider - manage report categories
- ReportGraphInfo, ReportGraphInfoProvider - manage graphs
- ReportInfo, ReportInfoProvider - manage reports
- ReportTableInfo, ReportTableInfoProvider - manage tables
- ReportValueInfo, ReportValueInfoProvider - manage values
- SavedGraphInfo, SavedGraphInfoProvider - manage saved graphs (for saved reports)
- SavedReportInfo, SavedReportInfoProvider - manage saved reports

The following chapters show how you can access the reports using the API.

## 52.8.2 Saving 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 user control CMSReporting \DisplayReport.ascx with ID set to "displayReportControl". The code behind will look like this:

**[C#]**

```
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(2006, 1, 1);

    dtp.Rows.Add(parameters);

    // Disable loading parameters from basic form
    displayReportControl.LoadFormParameters = false;

    // Choose report 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 occured or
    // SavedReportID of the successfully saved report
     displayReportControl.SaveReport();
}

protected override void Render(HtmlTextWriter writer)
{
    base.Render(writer);
}
```

## 52.8.3 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:

**[C#]**

```
using CMS.Reporting;

...

        // Saved report id
        private int mSavedReportId = 10;

        // Saved report info object
        SavedReportInfo sri = null;

        // Get saved report info by id
        sri = SavedReportInfoProvider.GetSavedReportInfo(mSavedReportId);

        if (sri != null)
        {
            // Set saved report HTML to the literal control text
            ltrReportHTML.Text = sri.SavedReportHTML;
        }
```

# Part XX

# 53 Module RSS feeds

## 53.1 Setting up a RSS Feed

Kentico CMS allows you to publish content using a RSS 2.0 feed. The default installation contains a simple **CMSPages\NewsRss.aspx** page that 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 figure shows the code of the rss.aspx page:

```
<%@ 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 a RSS Feed 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 rss.aspx file except for the <%@ Page %> directive.

3. Change the following properties of the CMSRepeater control:
   - SelectNodesOrderBy="DocumentModifiedWhen DESC"
   - SelectNodesClassName="cms.article"
   TransformationName="cms.article.rssitem"
   SelectedItemTransformationName="cms.article.rssitem"
   SelectNodesPath="/%"
   SelectNodesWhere=""

4. Add the same line of code as used in the rss.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.

# Part XX

# 54 Module Tags

## 54.1 Tags module overview

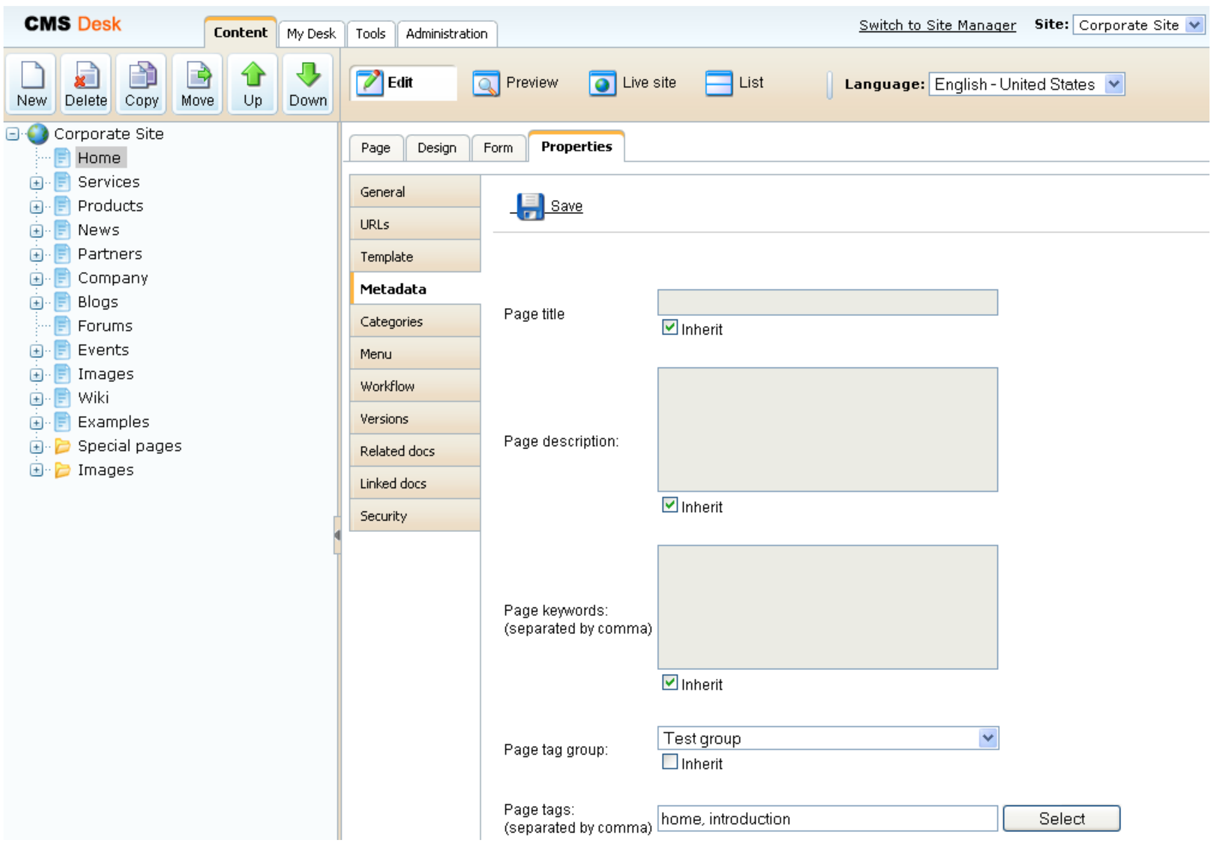
The Tags module allows you to tag documents with certain key words. A list can then be displayed, listing all documents tagged with a particular key word. The module also comes with the Tag cloud web part that can display a group of tags. The tags are displayed in a form of links. Clicking a link in the tag cloud will display a list of documents tagged with the clicked tag.

## 54.2 Tagging documents

The following example shows how to tag a document:

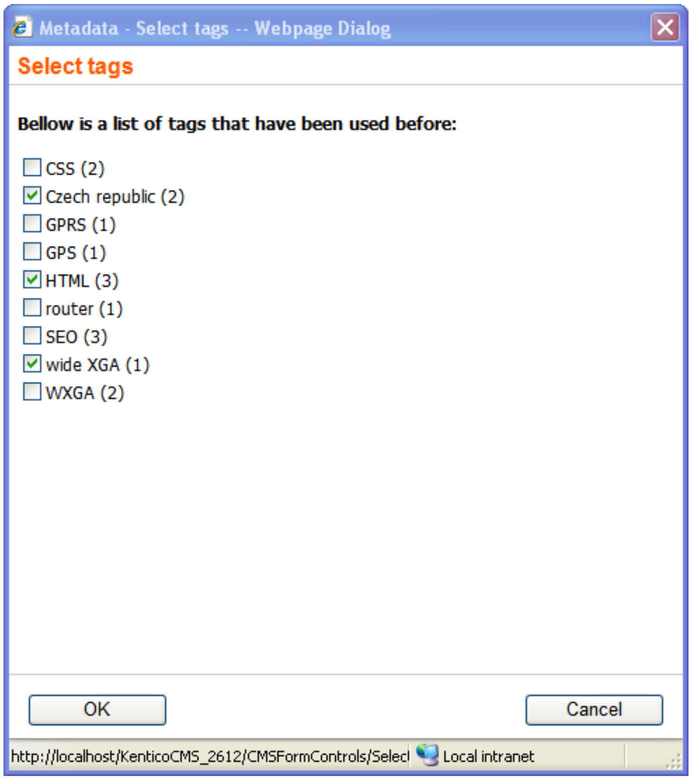
1) Go to **CMS Desk**, switch to **Edit** mode and select the document that you wish to tag from the content tree.

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.

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*

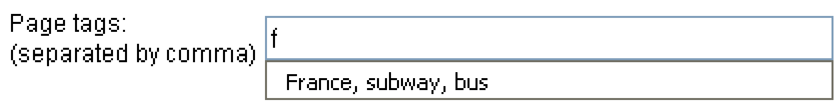
In case that you are entering a tag consisting of more than one word, you should enclose it within quotation marks. Multiple long tags can also be entered and can be also divided by both blank spaces and commas:

*"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:

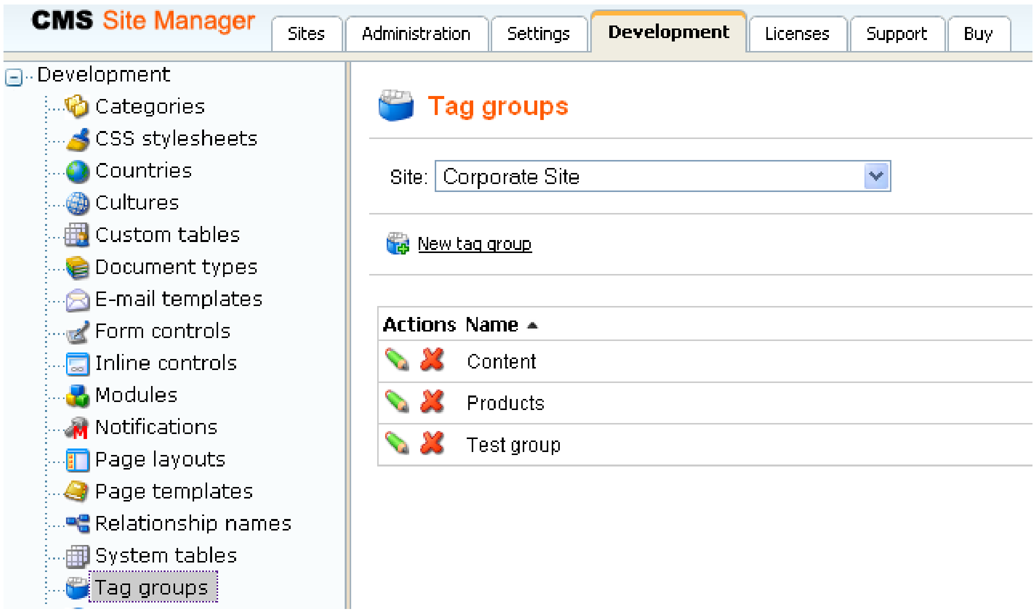
*"tag@1", "tag#2", "long, strange: tag@#$"*

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



## 54.3 Managing tag groups

Tags are divided into tag groups. These are topic-related groups of tags. Each tag group is related to a site, there are no global tag groups. Tag groups can be created and managed in **Site Manager -> Administration -> Tag groups**. You can **Edit** ( ) or **Delete** ( ) the listed tag groups in this section.

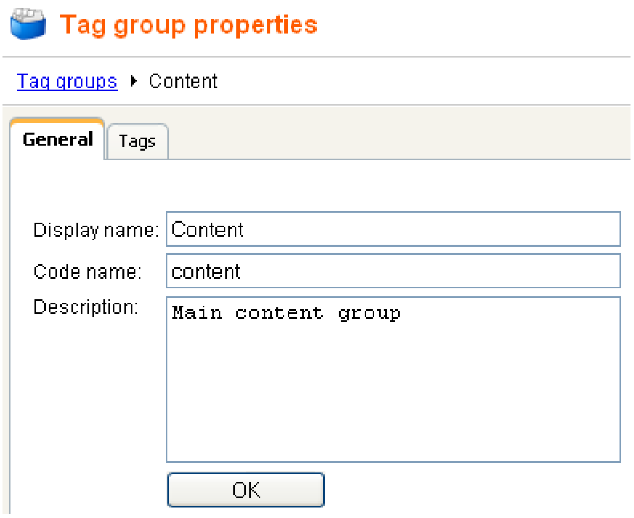



New tag groups can be created by clicking the **New tag group** link. The following properties will have to be entered:

- **Display name** - display name of the tag group
- **Code name** - code name of the tag group
- **Description** - description text of the tag group

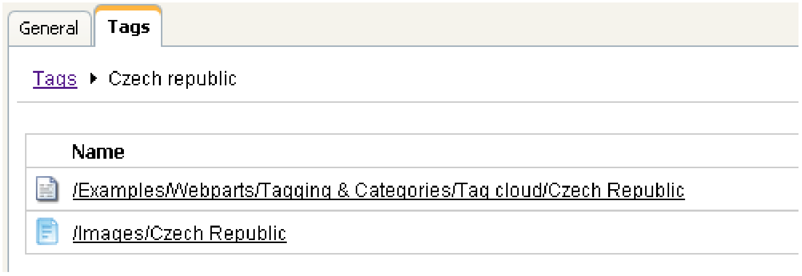
After clicking **OK**, the tag group will be created and become visible in the list.

If you choose to **Edit** some of the tag groups, its properties can be changed on the **General** tab.




The **Tags** tab displays a list of all tags in the selected group:

If you click the icon next to a tag, a list of links to documents tagged with the tag will be displayed.




## 54.4 Using the Tag cloud web part

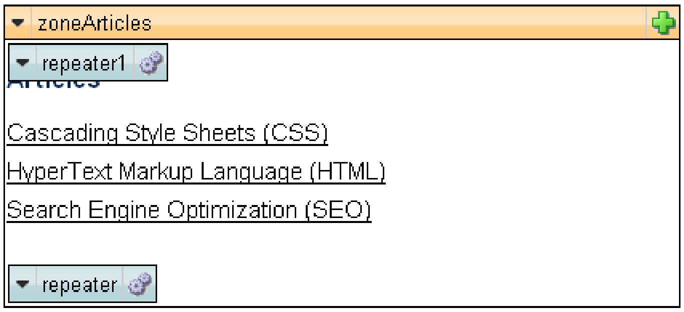
The Tag cloud web part is used to display tags that are associated with the current document. Tags are displayed in a form of links. Tags with higher occurrence are displayed in higher font size. If the site visitor clicks some of the links, a list of all documents tagged with the clicked tag will be displayed.

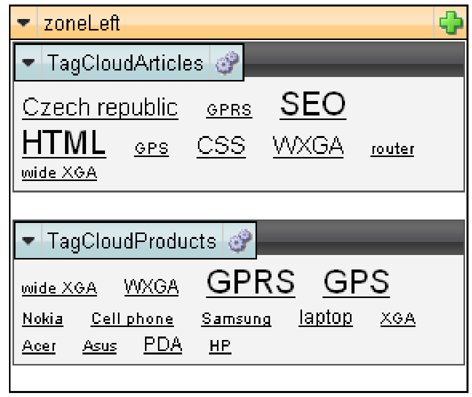

The web part has the following specific properties:

| | |
|---|---|
| Tag group name | Name of the tag group from that the tags will be displayed. |
| Document list URL | URL of the page on that the documents list will be displayed after clicking some of the tags. |
| Query string parameter name | Name of the parameter via that 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. |
| Select top N documents | Number of documents that will be displayed. |
| ORDER BY expression | ORDER BY 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 sample Corporate Site 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 via this parameter.

**Repeater**:

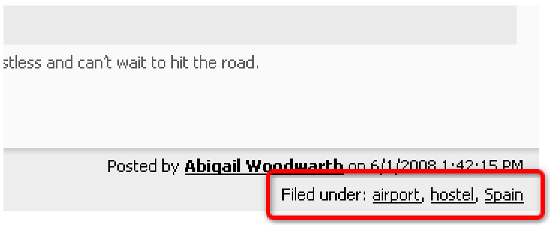
- Set the value of the **Path** parameter to the location in the content tree where the documents are stored.
- Set the value of the **Document types** parameter to the document type(s) that is (are) to be displayed.
- Select the transformations that you want to use for the **Transformation** and **Selected 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.



# Part XX

# 55 Module Time zones

## 55.1 Overview

The Time zones module enables configuration of time zone for the physical location of the server, for particular web sites and even for particular users.

A typical example of such use is displaying the time of forum post when you have a global community – while the server may be located in New York (GMT -5:00), the visitor coming from Paris (GMT +1:00) may see her new posts were added at 8am, while she would expect seeing 2pm according to her current time.

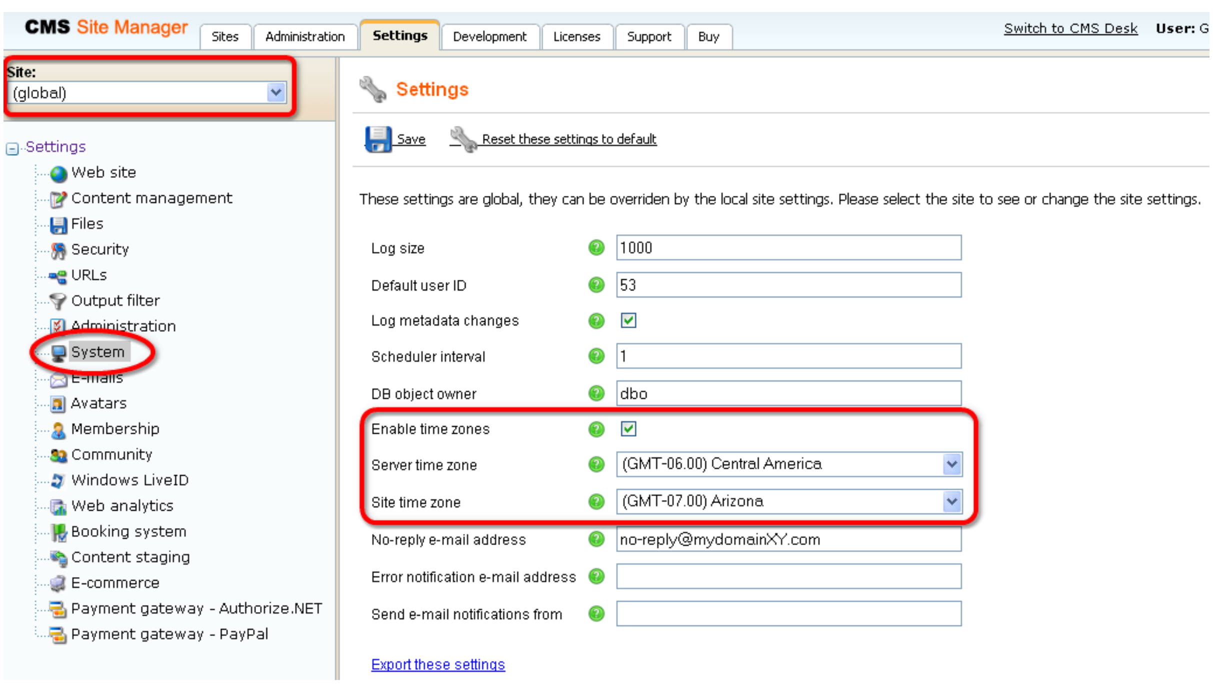
Another example is a web site of a global company that runs on a server in New York, but contains content for a French office. In this case, French visitor may wonder why the current time displayed by the server is 8am while it's 2pm in Paris. That's when you use the built-in support for multiple time zones.

Time zones are currently supported in all modules. The administration interface (CMS Desk and CMS Site Manager) does not reflect the time zone settings.

## 55.2 Enabling the module

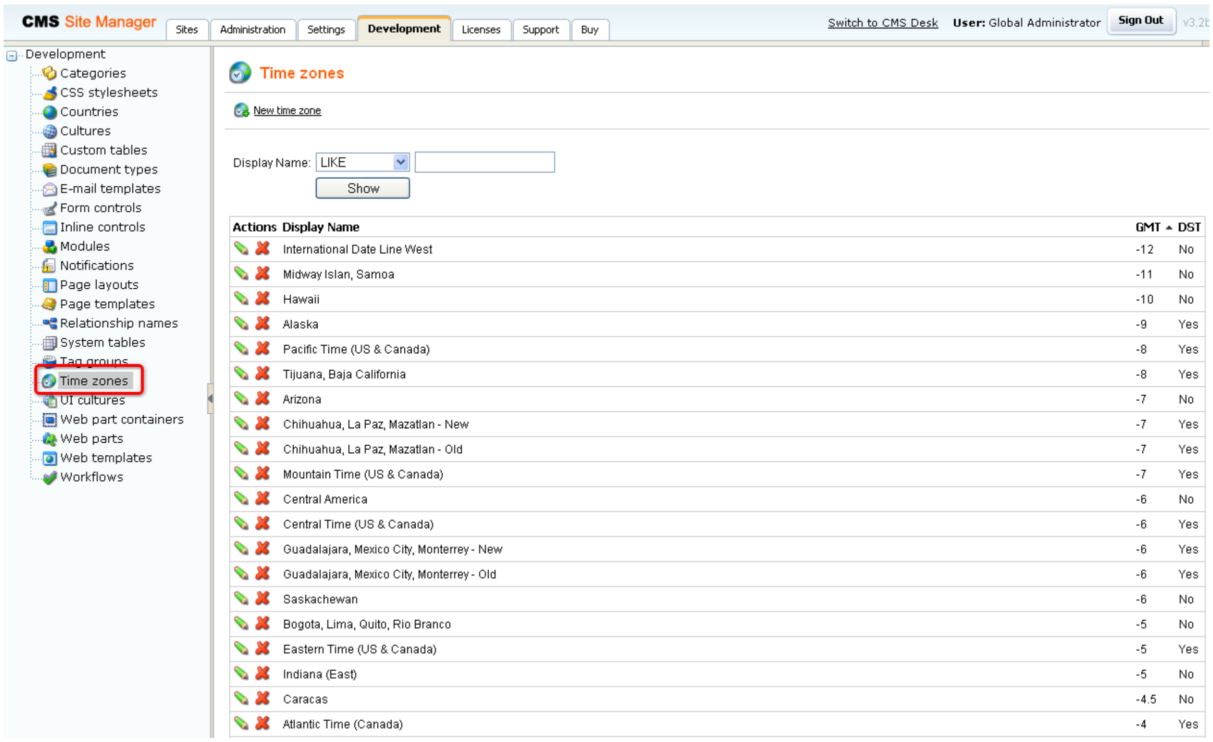
To enable the Time zones module, go to **Site Manager -> Settings -> System** and check the **Enable time zones** check-box. Below it, you can find the following two drop-down list:

- **Server time zone** - time zone of the physical server location
- **Site time zone** - default time zone of the web site 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



## 55.3 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.

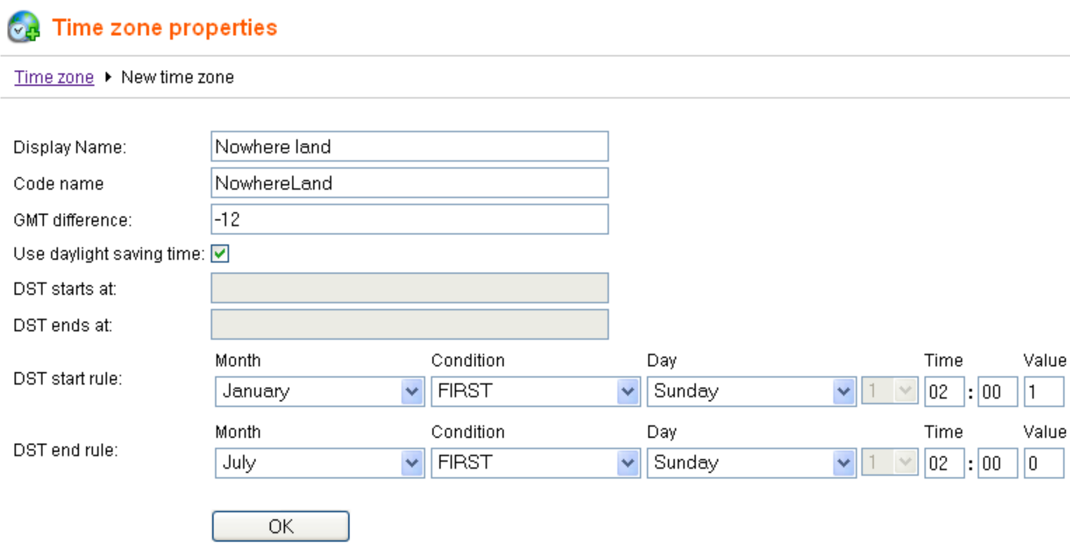
Time zones in the list can be **Edited** ( ) or **Deleted** ( ). You can also create new time zones by clicking the **New time zone** link.

### Creating a new time zone

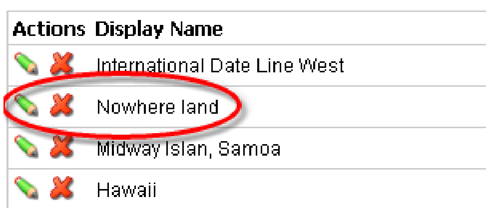
In the following example, you will learn how to create a new time zone:

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 the new time zone present among the records.

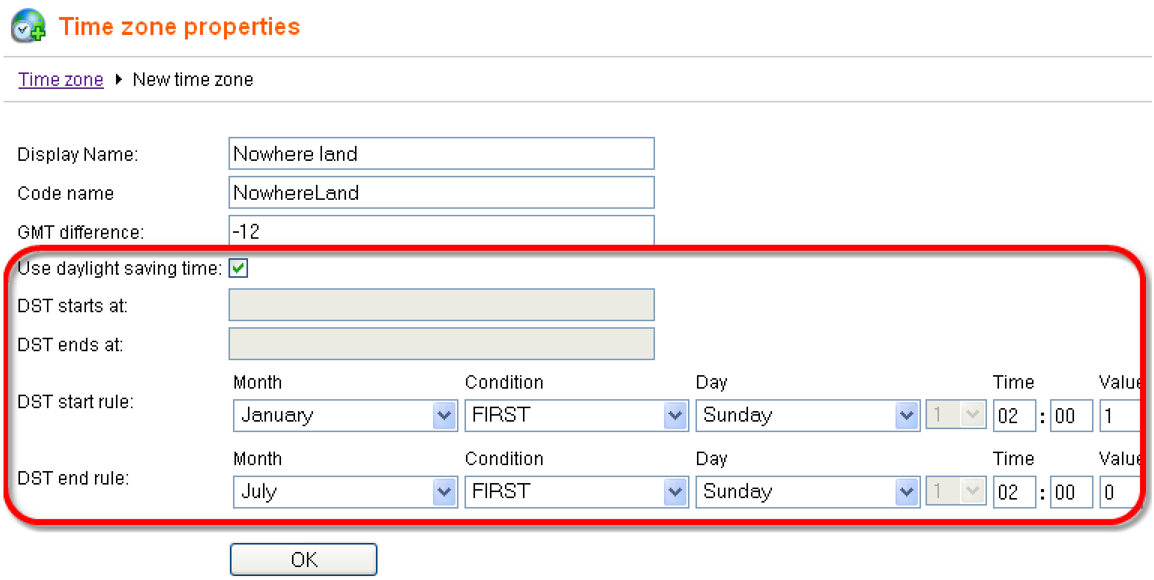



## 55.4 Daylight saving time

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 an one hour's advance in 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 the time zone to use the DST.

2) Now when you have the DST enabled, you have to set the **DST start rule** and **DST end rule** for the current time zone. First, select the month in that 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 will the change 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.

## 55.5 Use in web parts

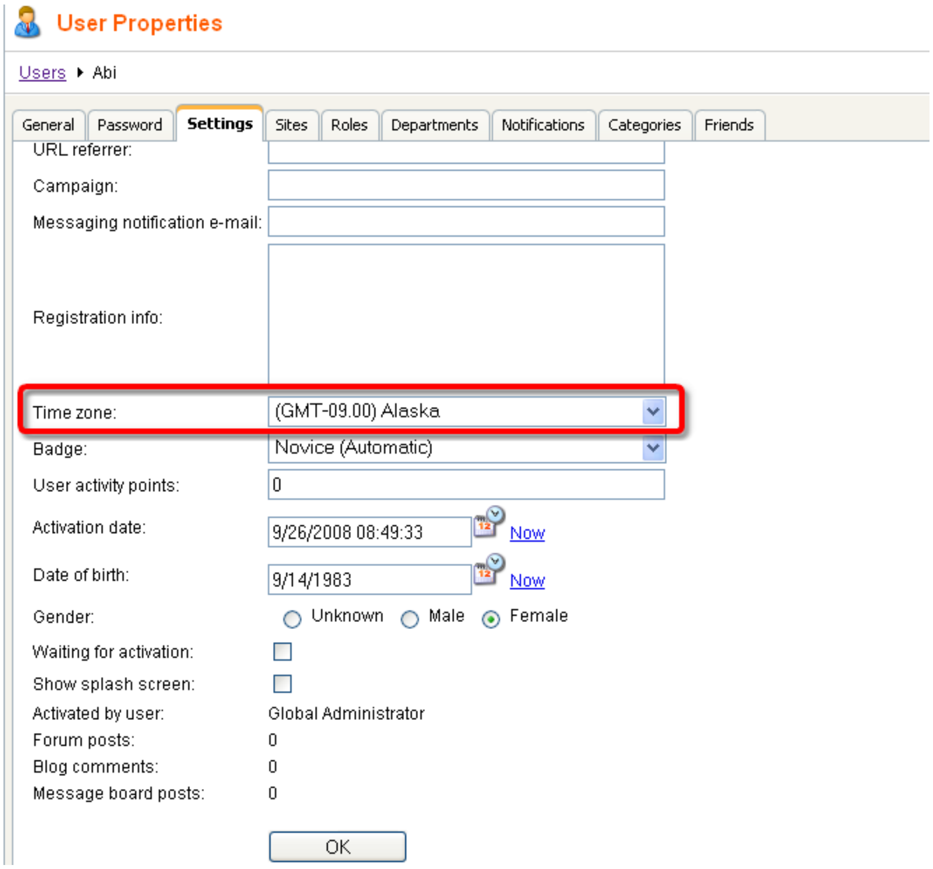
As you may have already noticed, all web parts have the **Time zones** section in their **web part properties**, where the applied time zone can be set. The section contains the following two properties:

| | |
|---|---|
| **TimeZone type** | **Inherit** - inherits the time zone setting from the Page placeholder in which it is displayed (typically the one on the master page)<br>**Server** - server time zone settings will be used by the web part<br>**WebSite** - web site time zone settings will be used by the web part<br>**User** - user's time zone settings will be used by the web part<br>**Custom** - some other time zone will be used based on the selection done by the Custom time zone property |
| **Custom time zone** | If you select some of the time zones, the selected time zone will be used by the web part, regardless of the user's or web site time zone settings. |

In 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 here.

## 55.6 Setting user's time zone

Each user can have her own time zone settings. Where applicable, this time zone settings is used instead of the web site's default time zone. 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.



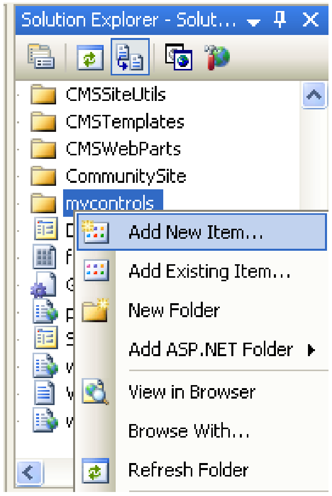

## 55.7 Displaying correct time in your code

The following methods can be used in transformation code to display correct time according to time zone settings.

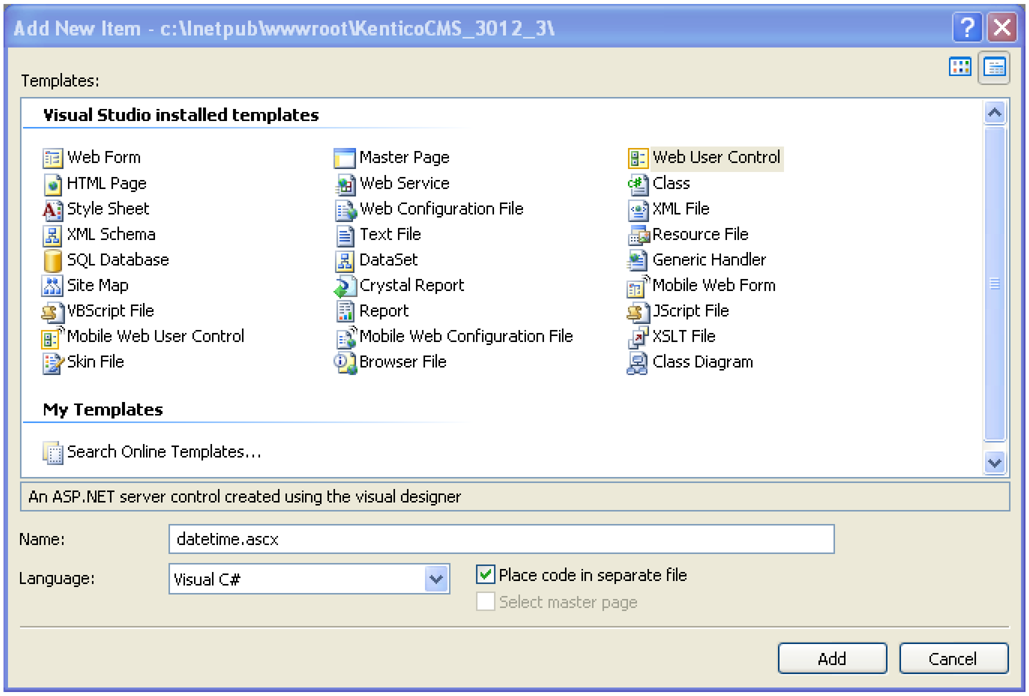
| | |
|---|---|
| <%# GetDateTime(DateTime.Now) %> | Returns date-time value according to web part time zone settings. |
| <%# GetSiteDateTime(DateTime.Now) %> | Returns date-time according to site time zone. |
| <%# GetUserDateTime(DateTime.Now) %> | Returns date-time according to current user time zone. |
| <%# GetCustomDateTime(DateTime.Now, "GreenwichMeanTime") %> | Returns date-time according to 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 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.


Solution Explorer - Solut...
CMSSiteUtils
CMSTemplates
CMSWebParts
CommunitySite
mycontrols
Add New Item...
Add Existing Item...
New Folder
Add ASP.NET Folder
View in Browser
Browse With...
Refresh Folder

2) Create a new WebUserControl in the mycontrols folder, name it datetime.ascx.




3) Add the following code to the PageLoad method of the user control.

```
Label1.Text = CMS.CMSHelper.CMSContext.ConvertDateTime(DateTime.Now, this
).ToString();
```

4) Add the General -> User control web part somewhere to some page of your web site. Set the following properties of the web part:

- **User control virtual path property:** ~/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 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.

# Part XX

# 56 Module User contributions (Wiki)

## 56.1 User contributions overview

The **User contributions** module allows you to create content editing interface for site members. It means that chosen web site visitors can create, edit and delete content, even if they are not editors and cannot access Kentico CMS Desk.

There are several scenarios where you can use this module, for example:

**1. Community news**

You can create a list of news an allow community members to add news without going to CMS Desk.

**2. Partner directory**

Your business partners can manage their profile on your web site 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** button.
- **Edit contribution** - this web part allows you to edit an existing document.

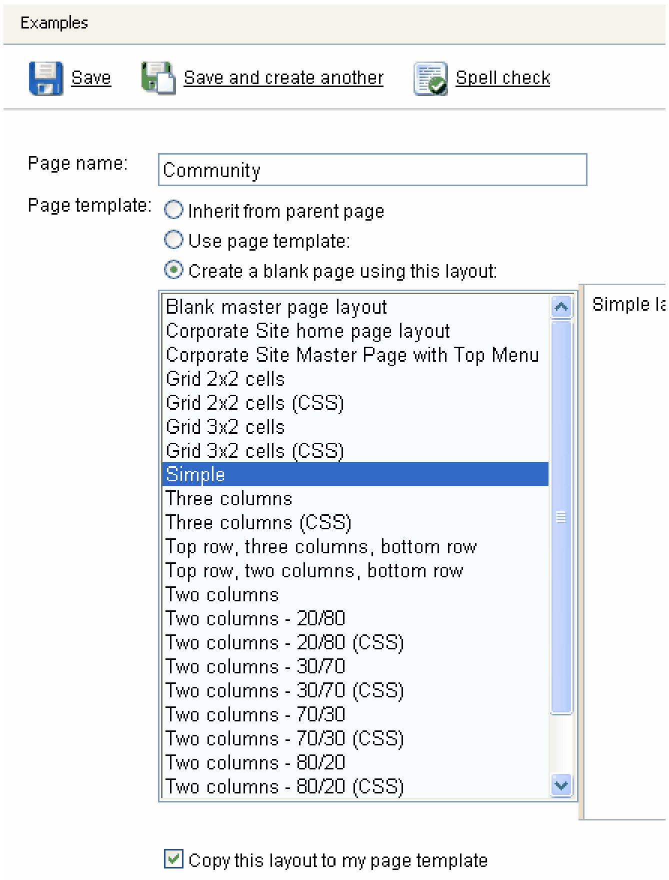
You can find the description of both web parts in **Kentico CMS Web Parts and Controls Reference**.

## 56.2 Example: Publishing community news

In this example, we will create a new section for publishing of community news. Every registered site member can create new news items and edit/delete her previously posted news. The example assumes you are using the sample Corporate Site.

## Creating the community news section

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:




Click **Save**. Switch to the **Design** mode of the newly create page and add the **Listings and Viewers/ Repeater** web part. Set the following properties:

- **ID:** repeaterNews
- **Document types:** cms.news
- **Transformation:** CMS.News.preview
- **Selected item transformation:** CMS.News.default

Click **OK**.

## Adding the New document button

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.*

- **Display list of documents:** no (unchecked)
  *- this ensures that the web part displays only the **New document** link, without displaying the list of documents.*
- **Allowed new document types:** CMS.News
  *- this means that the users will be allowed to create only news items under this section.*
- **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 web site (not to the administration interface).*

Leave the other values as they are by default.

Click **OK**.

## Adding the editing support

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

Click **OK**.

## Creating a testing user

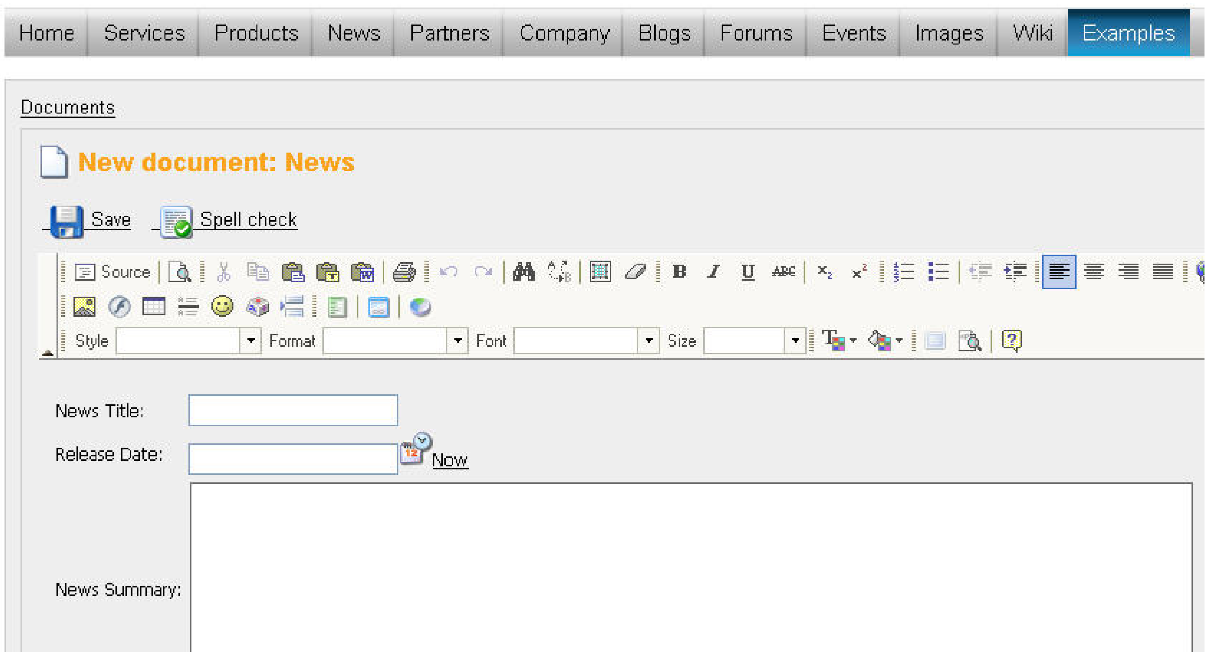
Go to **Administration -> Users** and create a new user with user name **test1**. In the **Is editor** field, choose no (uncheck the box) - this is a common site member without access to Kentico CMS Desk.

## Creating your first news item

Sign out and go to the **Examples -> Community** section. You will see a blank page since the user contributions are now only enabled 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:

Click the **New document** link and choose to create a **News** document:

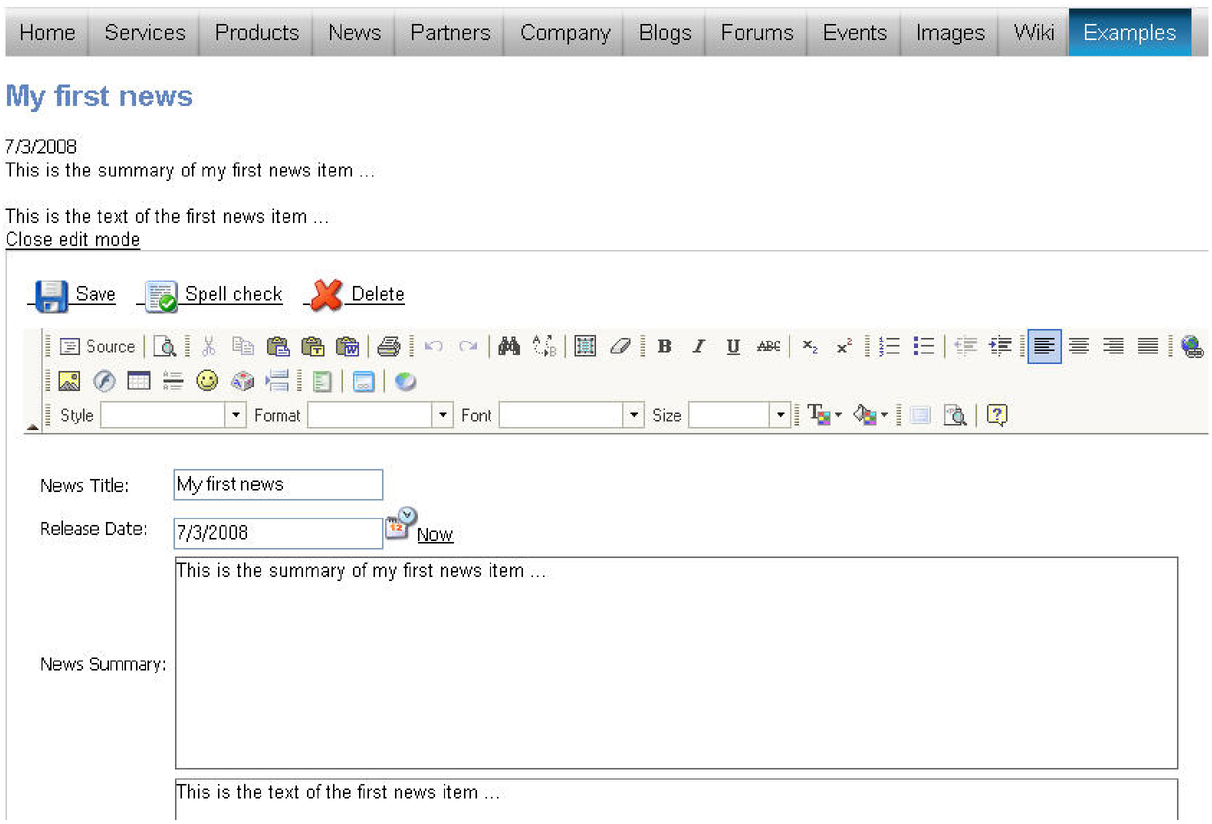

You are now displayed with the news editing form:




Enter some content and click **Save**. Click the **Documents** link at the top. You will see the **Community** page with your first news item:

Now click the news item to see the details page. You will see the **Edit/Delete** buttons:


Click **Edit** and you will be displayed with editing form where you can edit the content:




Make some modifications and click **Save**. Click **Close edit mode** to see the updated page.

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.

## 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 some site manager. You can find more details on workflow configuration in chapter Workflow overview.

# 56.3 Example: Editing partner profile

In this example, we will create a list of partners where partners can edit their profile on the My profile page after they sign in. This example assumes you're using the Corporate Site sample web site.

## Creating the partner document type

Before we create the page, we need to create a new document type **Partner**. Sign in as administrator to **CMS Site Manager** and go to **Development -> Document types**.

Click **New document type** and enter the following values:

**Step 1: General**
- **Document type display name:** Partner
- **Document type code name:** custom.Partner

**Step 2: Data type**
Choose the option "The document type has custom fields" and set:
- **Table name:** custom_Partner
- **Primary key name:** PartnerID

**Step 3: Fields**
Create the following fields:

- **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)

**Step 4: Additional Settings**

- **Document name source:** PartnerName

**Step 5: Parent types**

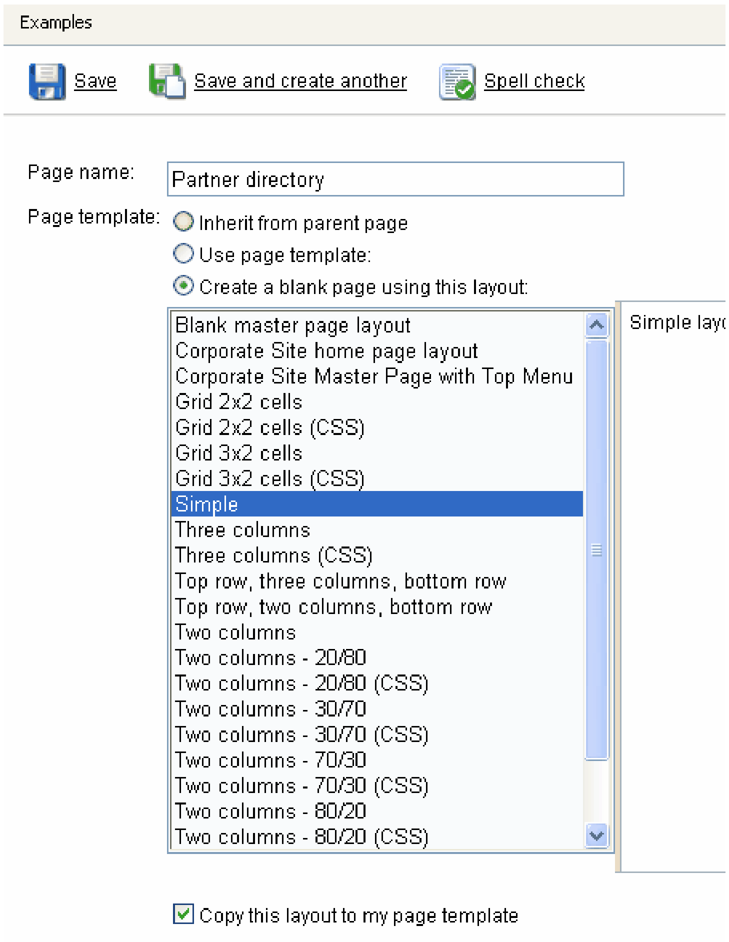
Choose only **Page (menu item)**.

**Step 6: Sites**

Choose your current web site.

## Creating the partner list

Go to **CMS Desk -> Content**, click **Examples** and click **New**. Choose to create a new **Page (menu item)** document. Call the page **Partner directory** and choose the **Create a blank page using this layout** option. Choose the **Simple layout** and click **Save**.




Switch to the **Design mode** 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: we will use the default transformation generated by the system, you can later modify the transformation so that it meets your design and layout requirements.

Click **OK**. Switch to the **Page** view and you will see an empty page now.

## Creating the partner user account

Go to **CMS Desk -> Administration -> Users** and create a new user with user name **AAAWebDesign** . Set the **Is editor** property to no (unchecked).

## Creating the partner profile document

Go to **CMS Desk -> Content**, click **/Examples/Partner directory** and click **New**. Choose to create a new **Partner**. Enter the following values:

- **Partner name:** AAA Web Design Inc.
- **Partner profile:** We provide web development services.

Click **Save**. Go to **Properties -> General** and set the **Document alias** to **AAAWebDesign** and click **Save**. We need to use the same alias as the user name since we 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>`.

When you see the page in the **Live site** mode now, you will see a page like this:


## Creating the partner profile editing page

Now we will create a page that will be available only for partners and it will allow them to edit their partner profile. Choose the **/Examples** document in the content tree and click **New**. Choose to create a new **Page (menu item)**. Call the page **My profile** and choose the **Create a blank page using this layout** option. Choose the **Simple layout** and click **Save**.

Switch to the **Design mode** of the newly created page and add the **Listings and viewers/Repeater** web part. It will display the current user's partner profile. Set the following properties 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 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>

Add the **User contributions/Edit contribution** web part. Set the following values:

- **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 current site visitor.*
- **Allow delete:** no (unchecked)

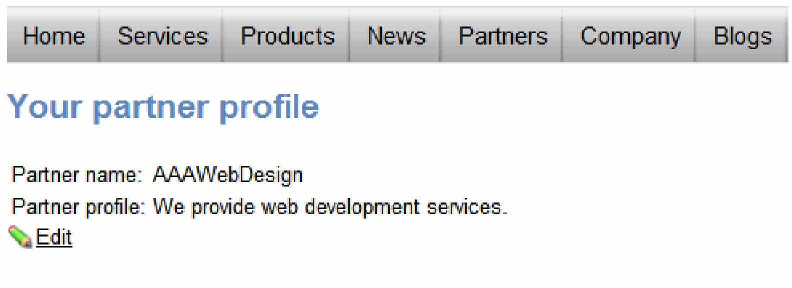
*- this option hides the Delete button since we do not want the partners to delete their profile accidentally.*

- **Allow editing by users:** Authenticated
  *- we want to allow editing only to authenticated users; at the same time, we do not require the user to be document owner.*

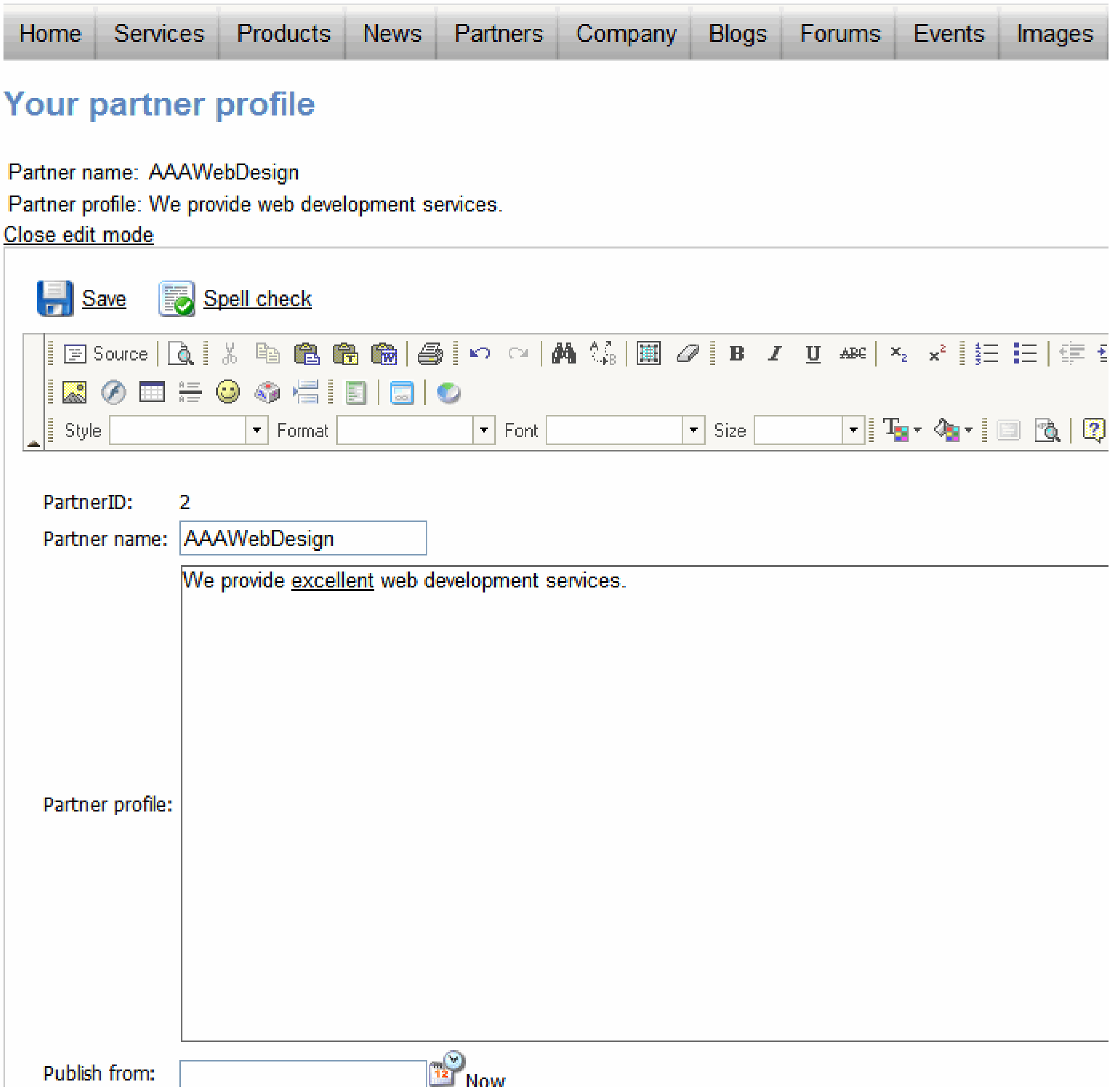
Go to the **Properties -> Security** dialog, set the **Requires authentication** value to **Yes** and click OK- we want to allow the access to this page only to authenticated partners so that we know their user name and can display their profile.

## Testing the profile editing page

Sign out and go to **Examples -> My profile**. You will be displayed with logon form. Sign in as user **AAAWebDesign**. You will see your partner profile with **Edit** button:



Click **Edit**, modify some values and click **Save**:




Go to **Examples -> Partner directory** and you will see the updated profile:

In this example, you have seen how you can allow users (partners) to edit a single document that matches their user name. It was a little different to the previous example (Example: Publishing community news), since in this case, we have created the document first and then we mapped it to the user by matching the document alias path and user name. The partner wasn't allowed to create new documents in this example and we didn't use the "document owner" security option.

**Tip 1: Simplifying the process of creating a new partner**

In cases like this, it's useful if you 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 (Managing users) and the partner profile document (Create document).

**Tip 2: Assigning multiple users to the same partner profile**

If the user name/document alias path mapping doesn't 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'll find your new custom field on the **Custom Fields** tab, not on the **General** tab as the default attributes.

## 56.4 Security

The user contributions web parts use the following properties to configure the security options:

- **Allow insert/edit/delete** - indicates if the appropriate buttons should be displayed.

- **Allow editing by users** - you can choose between:
  **- All** - any user who comes to the page with user contribution web part can use it to edit documents
  **- Authenticated** - any authenticated user (site member) can edit the documents
  **- Document owners** - only document owner can edit the documents

- **Check permissions** - if you choose this option, the users need to be granted with appropriate permissions to read/modify/create/delete documents using the User contributions web parts. See chapter Permissions for modules and documents for more details on document permissions.

## 56.5 User contributions and API

The user contributions 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 chapter Content management internals.

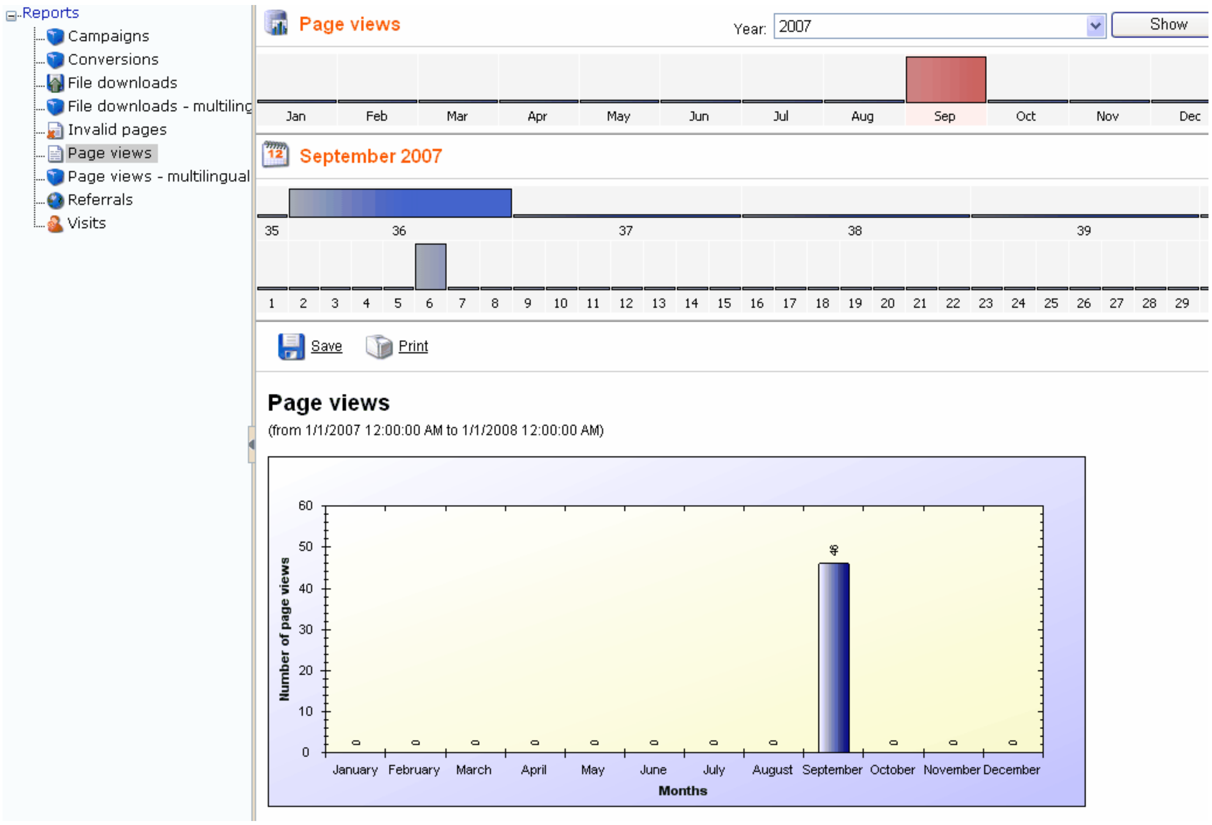
# Part XX

# 57 Module Web analytics

## 57.1 Web Analytics Overview

The Web Analytics module allows you to track and analyze web site visits, page views, file downloads and other metrics of the web site.

You can find the Web Analytics module in **CMS Desk -> Tools -> Web Analytics**:




When you select a report on the left, you can see the statistics on the right. You can choose the year, month, week and day. In the main window below, you can see the detailed reports with graphs and tables.

### 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 box **Enable Web Analytics**.

**Disk Permissions**

Please note that the Web Analytics module requires that your web application is granted with Modify permission on the disk, for the ~/app_data folder.

**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.

## 57.2 Available reports

The Web Analytics module comes with the following reports:

- **Browser types** - this report shows what types of browsers your web site 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 particular URL parameter to your web site, you can track the visit from particular campaign (typically from e-mail or banner campaigns). See Tracking marketing campaigns for more details.

- **Conversions** - this report helps you track the conversions on your web site. 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 higher number of visits.

- **File downloads** - this report summarizes the number of files downloaded by the web site 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).

- **Invalid pages** - this report summarizes the number of ASPX pages that were not found on the web site 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.

- **Referrals** - this report summarizes the number of visits coming from external web sites and the web sites from which the visitors come to your web site most often.

- **Visits** - this report summarizes the number of visits of your web site and the number of new versus returning visitors. The tracking is ensured by a cookie.

## 57.3 Tracking marketing campaigns

In many cases, tracking referring web site URLs may not be possible or reliable way of tracking your on-line marketing campaigns. Then, you may want to use the campaign tracking support in 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, we have added a parameter **campaign** with name of the campaign (banner1).

### How to configure campaign tracking

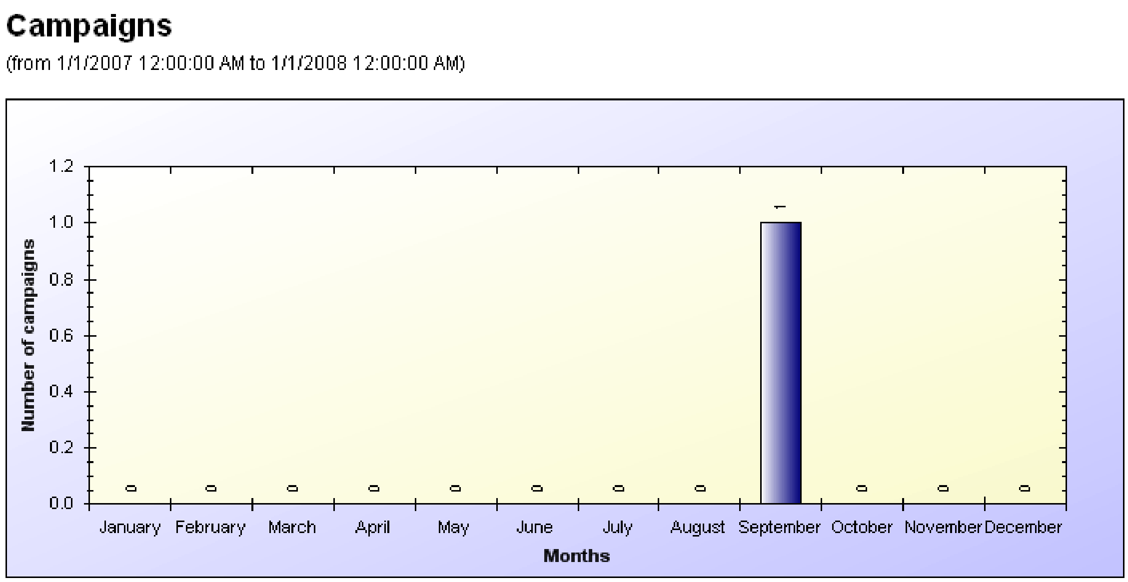
All you need to do is to specify the campaign tracking URL parameter name in **Site Manager -> Settings -> Web analytics**. Choose your web site and set the value of the **Campaign tracking URL parameter** key to: **campaign**

Then, modify the links to your web site so that they contain the paramter campaign like these:

```
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:




| Campaign | Views |
|---|---|
| banner1 | 1 |

## 57.4 Tracking conversions

The conversion tracking allows you to track the behavior of visitors on your web site and adjust the web site 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 web sites

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 web site, such as registration, sign up, etc. Then, you can see what changes made to the web site 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.Request.UserHostAddress))
{
   // 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 web site and/or campaign from which the user came:

**[C#]**

```
// the following values provide the information from user's cookie
CMSContext.CurrentUser.URLReferrer //web site 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-stored
values:
CMSContext.CurrentUser.UserURLReferrer //web site URL from which the visitor first came
CMSContext.CurrentUser.UserCampaign //campaign from which the visitor first came
```

## 57.5 Configuration options

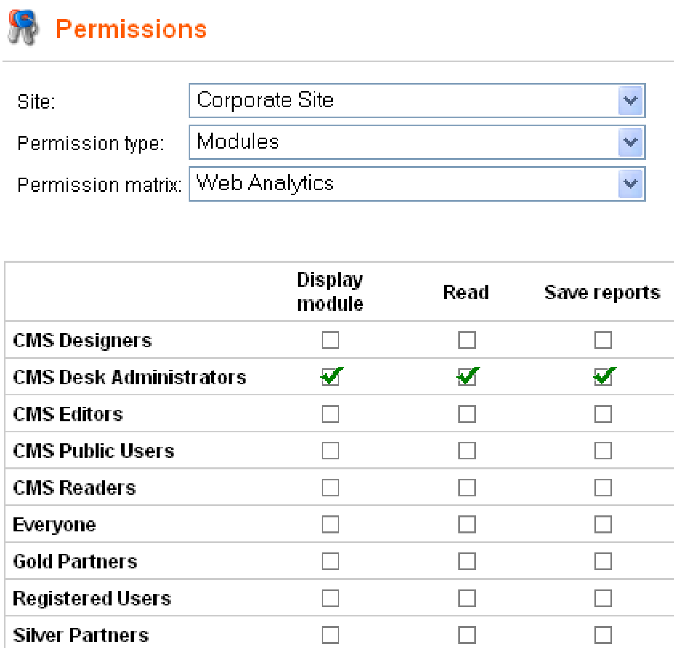
You can configure several options of the **Web Analytics** module in **Site Manager -> Settings -> Web Analytics** dialog:

| | |
|---|---|
| Campaign tracking URL parameter | The name of the URL parameter that is used to track campaigns. See Tracking marketing campaigns for more details. |
| Enable Web Analytics | Indicates if web events should be tracked and processed. |
| Exclude search engines | Indicates if search engine robots should be excluded from tracking. |
| 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 (;).<br><br>Example:<br><br>`.jpg;.gif`<br><br>**Please note:** it's necessary to include the dot in the extension name. |
| Excluded IP addresses | Client IP addresses whose access will not be tracked by the analytics module separated with a semicolon (;).<br><br>You can use it to exclude your own IP address so that your visits do not influence the results. |
| Excluded URLs | URLs of your web sites that will not be tracked. All underlying pages will not be tracked either. You can specify multiple URLs (or site sections) separated with a semicolon (;). |
| Track browser types, campaigns, conversions, etc. | Here you can configure which statistics should be tracked. Choosing only some of them may save database space and performance.<br><br>These options are only applied if the Enable Web Analytics box is checked. |

**Please note:** These options are applied at the moment when the event occurs. Later change will modify only future events.

## 57.6 Security

You can configure the access to the Reporting module in **CMS Desk -> Administration -> Permissions**. Choose the permission matrix for **Modules -> Web Analytics**.




| | Display module | Read | Save reports |
|---|---|---|---|
| CMS Designers | ☐ | ☐ | ☐ |
| CMS Desk Administrators | ☑ | ☑ | ☑ |
| CMS Editors | ☐ | ☐ | ☐ |
| CMS Public Users | ☐ | ☐ | ☐ |
| CMS Readers | ☐ | ☐ | ☐ |
| Everyone | ☐ | ☐ | ☐ |
| Gold Partners | ☐ | ☐ | ☐ |
| Registered Users | ☐ | ☐ | ☐ |
| Silver Partners | ☐ | ☐ | ☐ |

The following permissions can be given to the listed user roles:

- **Display module -** displays the module in the user interface
- **Read** - allows users to read Web Analytics reports
- **Save reports** - allows users to save reports. The saved reports can be viewed in the **Reporting** module

## 57.7 How it works

When an event (page view, file download, etc.) occurs, it's saved in the file in folder ~/app_data/CMSAnalytics. The format of the file 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 using the reports defined in the Reporting module.

The name of the scheduled task is **Analytics log processing** and you can control its status and settings in **Site Manager -> Administration -> Scheduled tasks**.

## 57.8 Creating custom reports

This example is describing 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

Example of logging of the button click event follows:

**[C#]**

```
protected void Button1_Click(object sender, EventArgs e)
{
  if (AnalyticsHelper.AnalyticsEnabled(CMSContext.CurrentSiteName))
  {
    HitLogProvider.LogHit("buttonclicked", CMSContext.CurrentSiteName, null,
CMSContext.CurrentUser.UserName, 0);
  }
}
```

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 DB 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. Next step is creation of reports which display statistics in graph or table form.

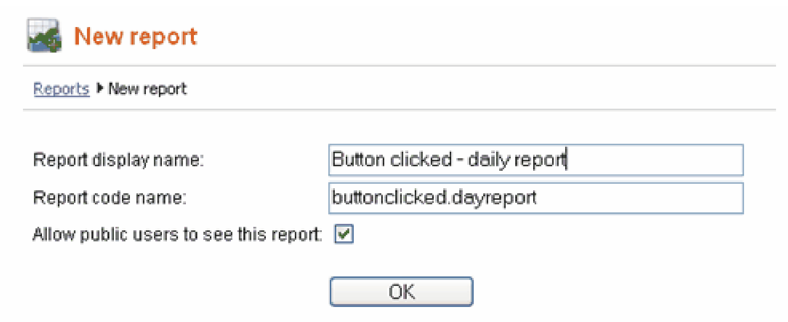
## Creating reports

Go to CMSDesk -> Tools -> Reporting and edit 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 specific form:

- <statisticscodename>.yearreport
- <statisticscodename>.monthreport
- <statisticscodename>.dayreport
- <statisticscodename>.hourreport

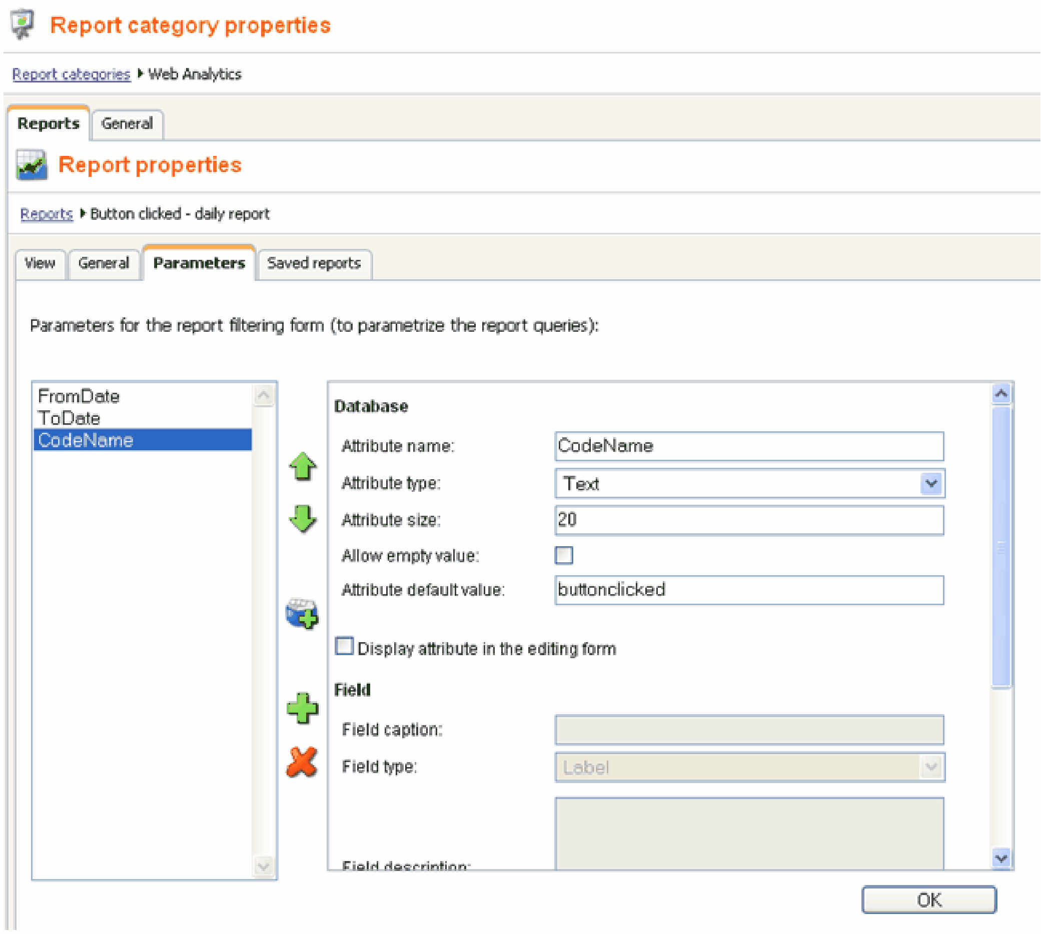
In our example <statisticscodename> is buttonclicked, as defined above.

Now we will create daily report for the new statistics.



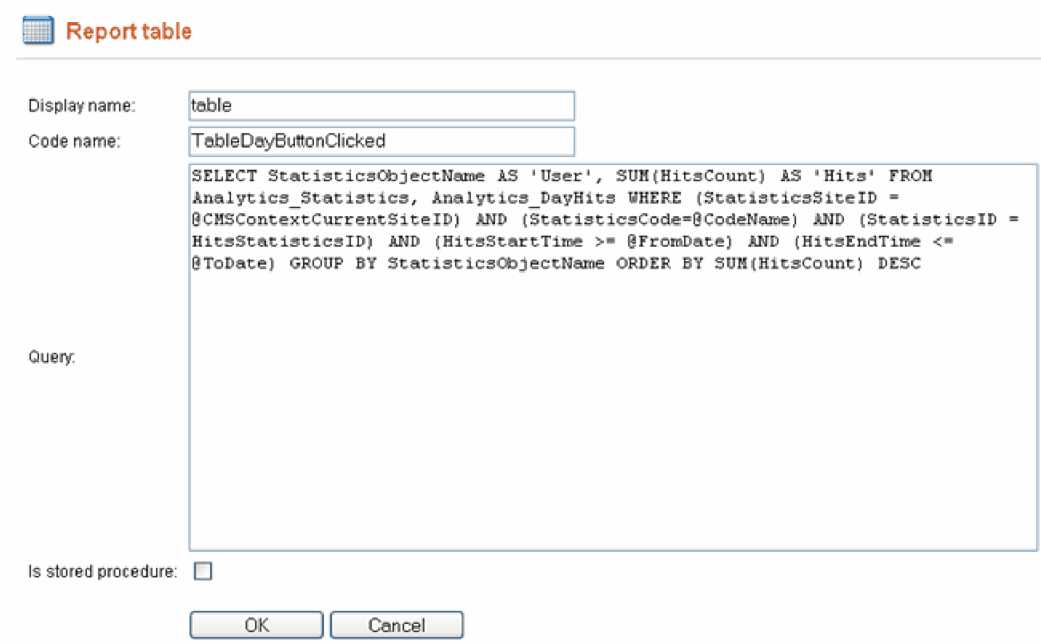
Go to Parameters section and create three essential parameters which are used in queries:

- FromDate – Date and Time attribute type
- ToDate - Date and Time attribute type
- CodeName – Text attribute type, statistics code name as attribute default value



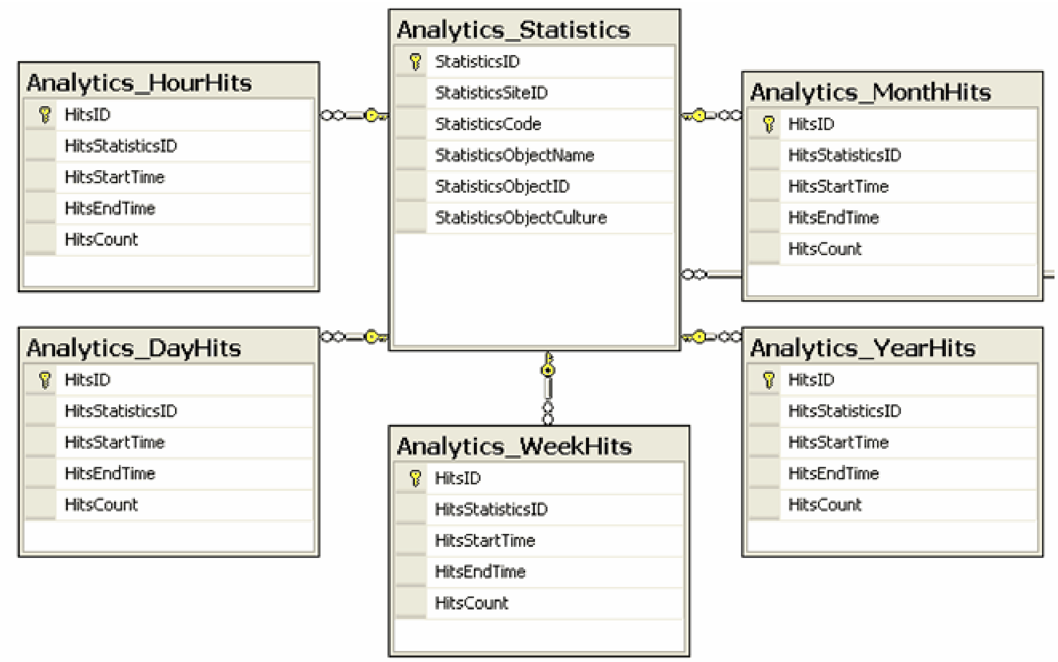

Disable the Display attribute in the editing form for all parameters.

Go to General section and click Add button near Tables to create report table:



There are six important DB tables used in web analytics module.

Table Analytics_Statistics contains general information about statistics, other five tables store statistics hit records for specific time period – Analytics_HourHits, Analytics_DayHits, Analytics_WeekHits, Analytics_MonthHits and Analytics_YearHits. As the report is daily report table Analytics_DayHits is used beside Analytics_Statistics table. See the diagram below to understand DB structure of web analytics:

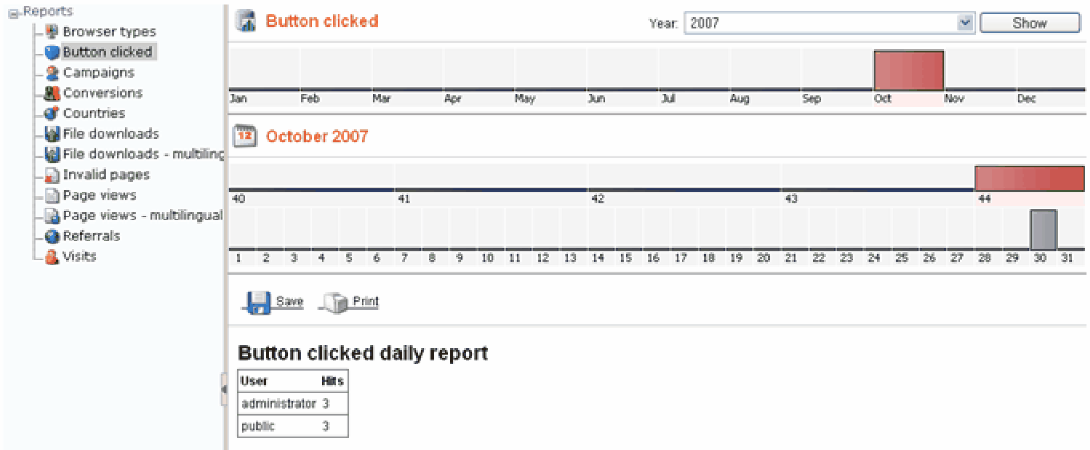



Now click OK and use Insert button near Tables to insert macro to the layout of the report. Click OK and daily report is ready now.

2. Other reports are similar, the only difference is in queries where you should use proper tables, like Analytics_YearHits in yearly report, Analytics_MonthHits in monthly report, etc.

3. Last thing you should do is to insert string into resources which is used in statistics list in CMSDesk -> Tools -> Web Analytics, form of the resource key have to be this – analytics_codename.<codename>, e.g. analytics_codename.buttonclicked for this example. You may also add specific icon for new statistics, icons are stored in '~\App_Themes\<used theme>\Images\CMSModules\WebAnalytics\Icons\', name should be in form <codename>.gif, if there are any '.' in codename, replace them with '_'.

And finally the result:



# Part XX

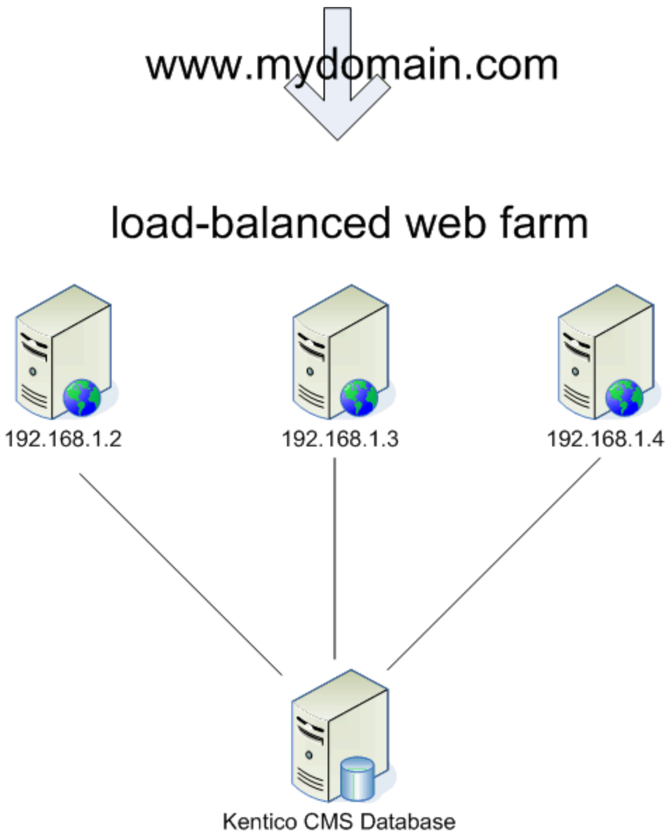
# 58 Module Web farm synchronization

## 58.1 Web farm support 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 that. However, if you're using web farm synchronization module, the module automatically creates a new synchronization task in the database and notifies the other servers so that they process their synchronization task.

**Using Kentico CMS in a web farm without Web farm synchronization module**

You can use Kentico CMS in 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**.

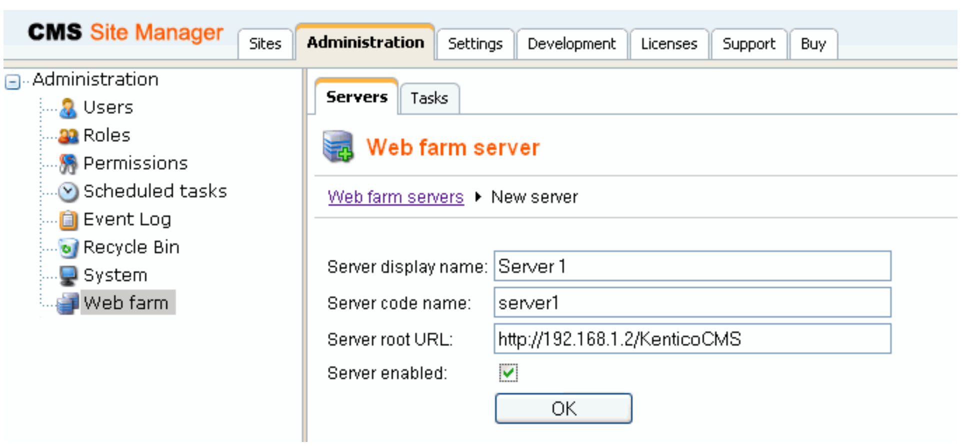
## 58.2 Defining web farm servers

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 server configuration file
- Server root URL: the URL of the root of the web site, such as

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 key into the /configuration/appSettings section:

```
<add key="CMSWebFarmServerName" value="server1"/>
<add key="CMSWebFarmEnabled" value="true"/>
```

where server1 is the appropriate server code name of the given server. Every server must contain only one such key with its own name.

If the synchronization doesn't work as expected, you can check the failed synchronization tasks on the **Tasks** tab. The processed tasks are automatically removed from the list, the failed tasks are displayed with 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 mydomain.com, and you access them internally through URLs like http://192.168.1.2, http://192.168.1.3, etc., you need to enter the license keys for mydomain.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.

# Part XX

# 59 API programming and Kentico CMS internals

## 59.1 API 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.

## 59.2 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 his/her preferences. Beside the standard UserInfo properties, it also provides the following ones:<br><br>**ViewMode** – the view mode in which the documents are displayed (edit, design, form, live site, preview).<br><br>**MenuItemViewMode** – the view mode used for page (menu item) documents (cms.menuitem type).<br><br>**PreferredUICultureInfo** – read-only property returning the current user's preferred UI culture (as CultureInfo object).<br><br>**PreferredCultureCode** – gets/sets user's preferred content culture. You can use this property to change the culture of the displayed content.<br><br>**PreferredUICultureCode** – gets/sets user's preferred UI culture. It's |

| | |
|---|---|
| | used mainly by Kentico CMS webparts, controls and in the administration interface.<br><br>**IsAuthorizedPerResource** – returns true if the current user is authorized with given permission for given module (resource).<br><br>**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.<br><br>**IsAuthorizedPerTreeNode** – returns true if the current user is authorized with given permission for given document (node).<br><br>**IsPublic** – returns true if the current user is the Public user account used for anonymous visitors. |
| WebApplicationVirtualPath | Returns the virtual path of the current web application. |
| 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.:<br>*/{0}/{1}*<br>applied on<br>*"/products/computers/ibm"*<br>results in:<br>*"/products/computers"* |

## 59.3 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. |

# 59.4 Managing documents

## 59.4.1 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

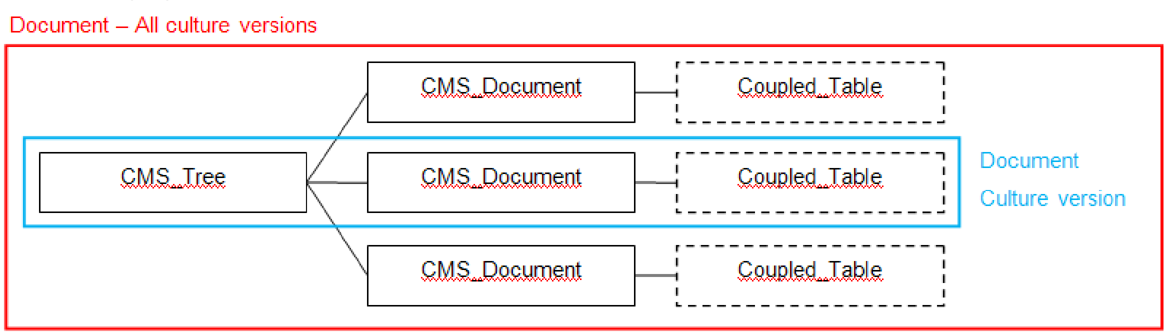
Document data is stored in several joined tables that are used for the tree structure, multilingual support and custom document fields. The document record consists of the following tables:

- **CMS_Tree** – a table with basic document data shared between different language versions of same document. This table determines the tree structure of the web site 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

## 59.4.2 Select document(s)

### Select single node by NodeID

**[C#]**

```
// 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#]**

```
// 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#]**

```
// 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#]**

```
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 ...
```

## 59.4.3 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
TreeNode parent = provider.SelectSingleNode(CMS.CMSHelper.CMSContext.CurrentSiteName,
"/News", "en-us");

// create a new tree node
TreeNode node = new TreeNode("cms.news", provider);

if (parent != null)
{
   // set document properties
   node.NodeName = newsName;
   node.NodeAlias = newsName;
   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);
}
```

## 59.4.4 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
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();
}
```

## 59.4.5 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
TreeNode node = provider.SelectSingleNode(CMS.CMSHelper.CMSContext.CurrentSiteName,
"/News/MyFirstNews", "en-us");

if (node != null)
{
   // Delete the document
   node.Delete();
}
```

## 59.4.6 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
TreeNode node = provider.SelectSingleNode(CMS.CMSHelper.CMSContext.CurrentSiteName,
"/Products/PDAs/ItemA", "en-us");
// Get target node under which the document will be copied
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);
}
```

## 59.4.7 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
TreeNode node = provider.SelectSingleNode(CMS.CMSHelper.CMSContext.CurrentSiteName,
"/Products/PDAs/ItemA", "en-us");

// Get target node under which the document will be moved
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);
}
```

## 59.4.8 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.

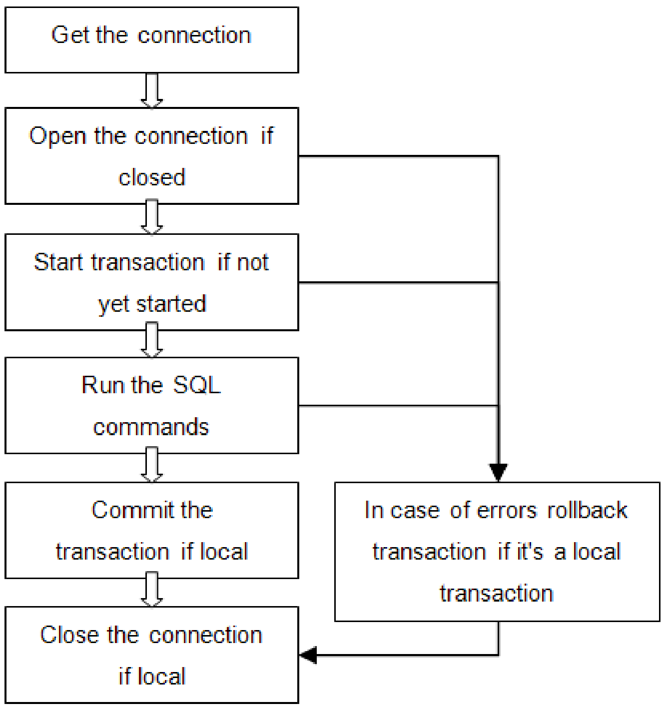
# Native transactions

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#]**

```
/// <summary>
/// Sample method
/// </summary>
/// <param name="conn">Data connection to use</param>
public void DoSomethingInTransaction(GeneralConnection conn)
{
  bool closeConnectionAtTheEnd = false;
  bool commitTransactionAtTheEnd = false;

  try
  {
    // Open connection is necessary
    if (!conn.DataConnection.IsOpen())
    {
      conn.DataConnection.Open();
      closeConnectionAtTheEnd = true;
    }
    // Start transaction if necessary
    if (!conn.DataConnection.IsTransaction())
    {
      conn.DataConnection.BeginTransaction();
      commitTransactionAtTheEnd = true;
    }


    // --- 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 if necessary
    if (commitTransactionAtTheEnd)
    {
      conn.DataConnection.CommitTransaction();
      commitTransactionAtTheEnd = false;
    }
  }
  catch (Exception ex)
  {
    // Rollback transaction if necessarry
    if (commitTransactionAtTheEnd)
    {
      conn.DataConnection.RollbackTransaction();
      commitTransactionAtTheEnd = false;
    }
    throw ex;
  }
  finally
  {
    // Close the connection if necessarry
    if (closeConnectionAtTheEnd)
    {
      conn.DataConnection.Close();
      closeConnectionAtTheEnd = false;
    }
  }
}
```

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.

### 59.4.9 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.";
            }
```

### 59.4.10 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);

                this.lblInfo.Text = "Link '" + node.NodeAliasPath + "' has been deleted.";
            }
```

## 59.5 Managing attachments and files (cms.file)

### 59.5.1 Managing attachments

he 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#]**

```
// 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#]**

```
// ... 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.

### 59.5.2 Managing files (cms.file)

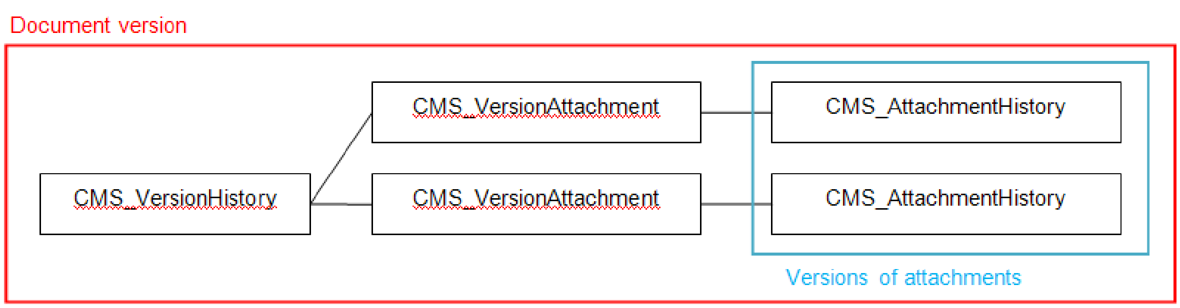
The File document type (cms.file) is actually just another document type and you can modify it using the same methods that you use for other document types. The actual file (uploaded image, Word document or other file type) is stored as an attachment in field **FileAttachment**.

## 59.6 Managing documents with workflow

### 59.6.1 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**.

### 59.6.2 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#]**

```
// 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#]**

```
// 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#]**

```
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.

## 59.6.3 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#]**

```
// 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#]**

```
// 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#]**

```
// 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);
```

## 59.6.4 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);
```

## 59.6.5 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);
            }
```

# 59.7 Security management

## 59.7.1 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.

## 59.7.2 Managing users

### Creating a new user

**[C#]**

```
// 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#]**

```
// 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#]**

```
// 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);
```

## 59.7.3 Setting user password

**[C#]**

```
using CMS.SiteProvider;

...

UserInfoProvider.SetPassword("testinguser", "newpassword");
```

## 59.7.4 Managing roles

### Creating a new role

**[C#]**

```
// 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#]**

```
// 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);
```

## 59.7.5 Managing user roles

### Getting users who belong to the given role

**[C#]**

```
// 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
   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#]**

```
// Get UserInfo of the specified name
UserInfo user = CMS.SiteProvider.UserInfoProvider.GetUserInfo("Alice");

if (user != null)
{
   // Get table with all user roles
   DataTable table = CMS.SiteProvider.UserInfoProvider.GetUserRoles(user.UserID);

   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.

## 59.7.6 Authenticating user

### Checking user name and password

**[C#]**

```
// UserInfo
UserInfo user = null;

user = CMS.SiteProvider.UserInfoProvider.AuthenticateUser("Alice", "*******",
CMS.CMSHelper.CMSContext.CurrentSiteName);

if (user != null)
{
   // Authentication was successful
}
```

## 59.7.7 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 user is granted with given permission for a document

**[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 user is granted with given permission for a module (resource)

**[C#]**

```
if (CMS.CMSHelper.CMSContext.CurrentUser.IsAuthorizedPerResource("CMS.Forums", "Modify"))
{
   // the current user is granted with permission "Modify" in the "CMS.Forums" module in
the current web site
}
```

### Checking if user is granted with given permission for a document type or custom table (class name)

**[C#]**

```
if (CMS.CMSHelper.CMSContext.CurrentUser.IsAuthorizedPerClassName("CMS.File", "Delete"))
{
   // the current user is granted with permission "Delete" for all documents of type
"CMS.File" in the current web site
}
```

# 59.8 Site management, import and export

## 59.8.1 Creating a new web site

The following example shows how you can create a new web site based on the Blank web site 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
.GetWebTemplateInfoByCodeName("BlankSite");

            // 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 been
created.", siteName);
                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";
        }
```

## 59.8.2 Import and export of the web site

You can export and import a web site 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 web site
  - siteDisplayName - display name of the newly created web site
  - siteDomain - domain name of the newly created web site
  - fullSourcePath - physical disk path of the package with exported site
  - websitePath - physical disk path of the web site root

- **`public static void ExportSite(string siteName, string fullExportFilePath, string websitePath, bool template)`**
  - siteName - code name of the web site to be exported
  - fullExportFilePath - physical disk path of the export package
  - websitePath - physical disk path of the web site 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).

## 59.8.3 Update web site properties

The following example shows you how to modify web site 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 dsiplay 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-US
                    SiteInfoProvider.AddDomainAlias(siteName, "newdomainalias", "en-US");

                    // 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: " +
ex.Message;
            }
            catch (Exception ex)
            {
                lblError.Text = "Error when modifying site properties.<br />Original
exception: " + ex.Message;
            }
        }
```

### 59.8.4 Delete web site

The following example shows how you can delete an existing web site:

**[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);
```

## 59.9 Custom Providers

### 59.9.1 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 (it's described in the Kentico CMS E-commerce guide)

You can also use global event handlers to customize the system behavior, such as document modifications, authentication process, etc.

### 59.9.2 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 2005.
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 web site project.
5) Add the reference do IDataConnectionLibrary located in the web site 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 web site, it should use the Custom Data Provider library now.
9) If everything works fine, you can modify the code of the Custom Data Provider as needed.

## 59.9.3 Custom E-mail Provider

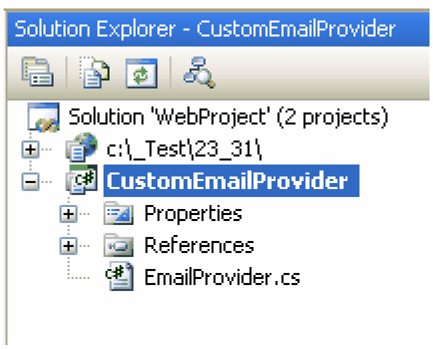
The custom e-mail provider allows you to use third-party e-mail component for sending e-mails or add custom action when an e-mail is sent (e.g. logging the sent e-mail to some file for auditing purposes). All e-mails sent by Kentico CMS and its modules will use your custom e-mail provider then.

### 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\2.x\CodeSamples\CustomEmailProvider to some development folder (not under the web project).

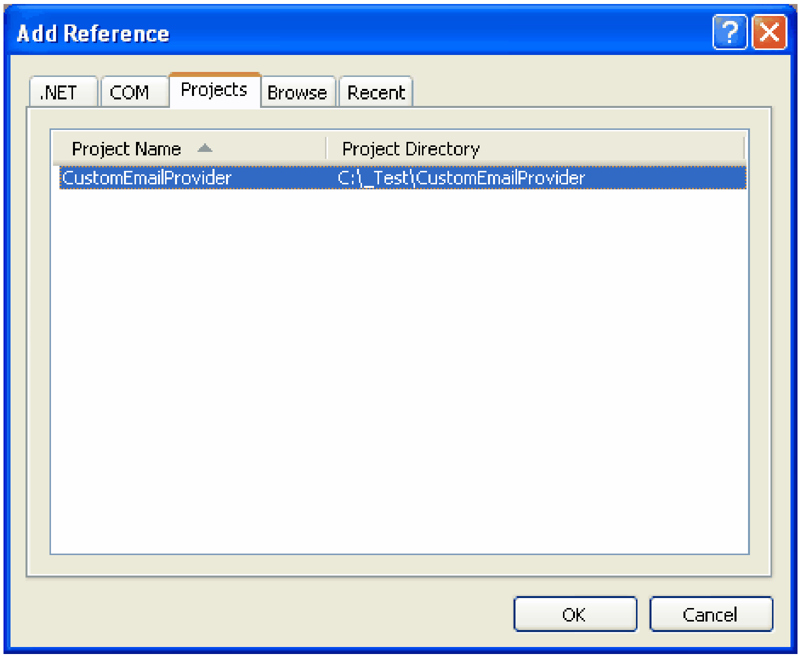
2. Open the **CMS** web project using the **WebProject.sln** file. Click **File -> Add -> Existing Project** and select the **CustomEmailProvider.csproj** file in the folder where you copied the **CustomEmailProvider** project. Your Solution Explorer window will look like this:




3. Unfold the **References** section of the **CustomEventHandler** 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

5. Unfold the **bin** folder in the **CMS web project**. Remove the reference to the **CMS.CustomEmailProvider.dll** library. 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 Search 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.

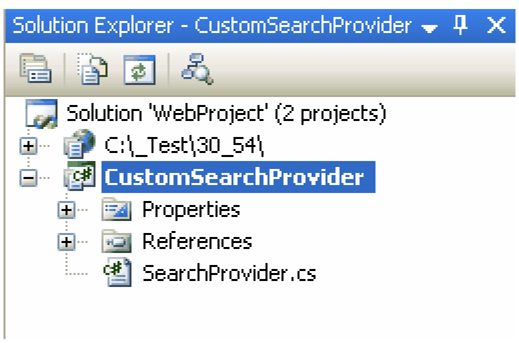
## 59.9.4 Custom Search Provider

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 modify the standard search results so that they display the document path in the results in format
**"> News > News 1"**
instead of
**"/News/News 1"**

Tip: This is an example. 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**.

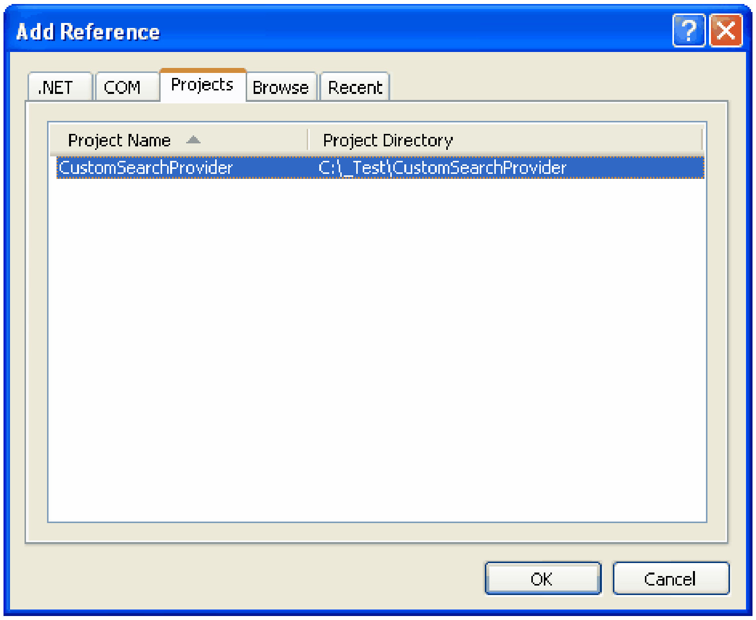
1. Copy the **CustomSearchProvider** project from Kentico CMS installation (typically C:\Program Files \KenticoCMS\2.x\CodeSamples\CustomSearchProvider to some development folder (not under the web project).
2. Open the **CMS** web project using the **WebProject.sln** file. Click **File -> Add -> Existing 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 **CustomEventHandler** 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**

5. Unfold the **bin** folder in the **CMS web project**. **Remove the CMS.CustomSearchProvider.dll** reference. 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 **CustomSearchProvider** library and place your code to the Search method. Enter the following code into the SearchProvider.Search method instead of the sample code:

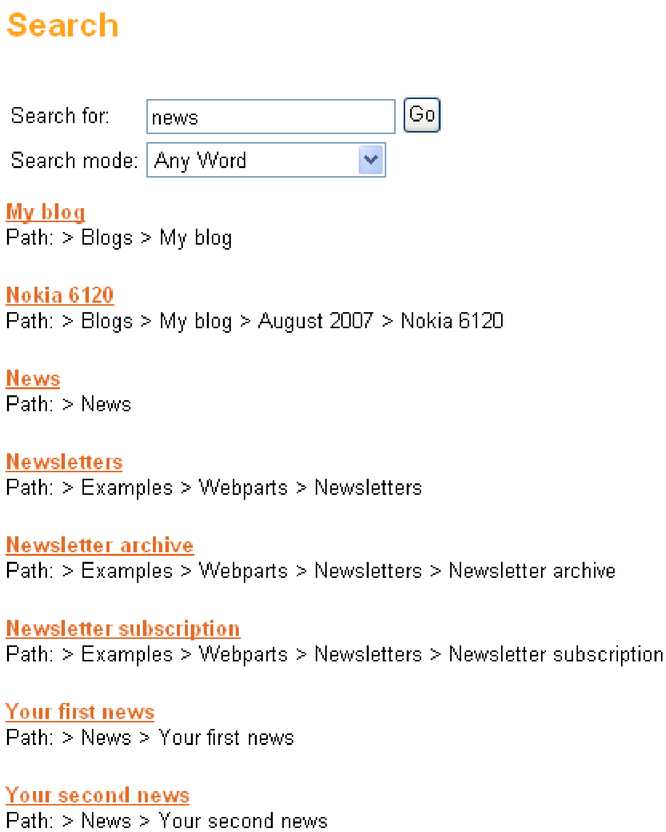
**[C#]**

```
CMS.SearchProviderSQL.SearchProvider standardSearchProvider = new
CMS.SearchProviderSQL.SearchProvider();
DataSet ds = standardSearchProvider.Search(siteName, searchNodePath, cultureCode,
searchExpression,
                searchMode, searchChildNodes, classNames, filterResultsByReadPermission,
searchOnlyPublished,
                whereCondition, orderBy);

if (ds.Tables.Count > 0)
{
   foreach (DataRow dr in ds.Tables[0].Rows)
   {
      dr["DocumentNamePath"] = ((string) dr["DocumentNamePath"]).Replace("/", " &gt; ");
   }
}
return ds;
```

7. Set the following value in 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. You will see the search results in the following format:



# 59.10 Data layer

## 59.10.1 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

## 59.10.2 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);
```

# 59.11 Global events and their handling

## 59.11.1 Event handling overview

Global events allow you to execute custom events when some CMS event occurs. For example: if a document is created in the CMS system, you can handle this event, get information on the new document and send its content by e-mail or use 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 your web project's web.config file contains the following value:

**<add key="CMSUseCustomHandlers" value="true" />**

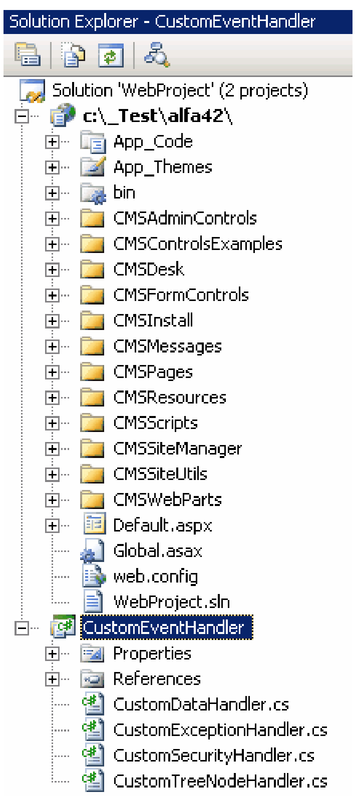
**Default event handler library**

The default 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" />**

## How to write your own event handler

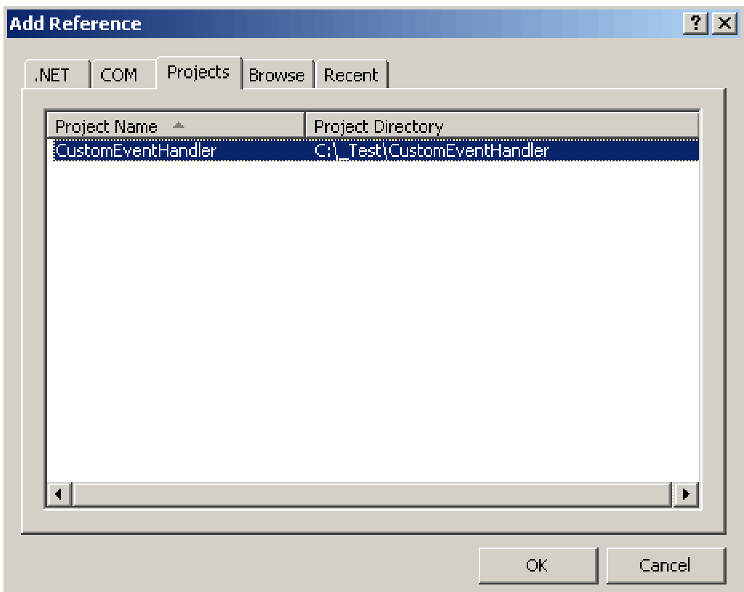
1. Copy the **CustomEventHandler** project from Kentico CMS installation (typically C:\Program Files \KenticoCMS\3.x\CodeSamples\CustomEventHandler to some development folder (not under the web project).
2. Open the **CMS** web project using the **WebProject.sln** file. Click **File -> Add -> Existing Project** and select the **CustomEventHandler.csproj** file in the folder where you copied the **CustomEventHandler** project. Your Solution Explorer window will look like this:



3. Remove the invalid link to file **CMS.pfx**. Unfold the **References** section of the **CustomEventHandler** project and delete references to libraries **CMSHelper** and **SiteProvider**.

4. Now you need to update the references. Unfold the **bin** folder in the 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 the **CustomEventHandler** project reference:




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.EmailEngine.dll
   - CMS.DataEngine.dll
   - CMS.WorkflowEngine.dll

Please note that you can also navigate to **bin** folder on the **Browse** tab and 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.EmailEngine.dll" "CMS.DataEngine.dll" "CMS.WorkflowEngine.dll"
```

7. Click **Build -> Rebuild solution**.

Now you can modify the **CustomEventHandler** library source code and add your own code to handle appropriate actions. 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.

## 59.11.2 Data handler (CustomDataHandler class)

This handler allows you to add custom actions to the following events:

- OnBeforeUpdate
- OnAfterUpdate
- OnBeforeInsert
- OnAfterInsert
- OnBeforeDelete
- OnAfterDelete

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 her 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
DataClass dataItem = (DataClass)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 you 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.

## 59.11.3 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.

## 59.11.4 Security handler (CustomSecurityHandler class)

The security handler allows you to integrate external user databases and modify the authentication and authorization process.

It handles the following events:

- OnAuthentication - the user tries to sign in with user name and password
- OnClassNameAuthorization - checking user's permissions for particular document type
- OnResourceAuthorization - checking user's permissions for particular module
- OnTreeNodeAuthorization - checking user's permissions for particular document
- OnFilterDataSetByPermissions - filtering a DataSet with documents based on permissions or custom personalization rules

### Example

In the following example, you will learn how to integrate an external user authentication using the custom security handler. For simplicity, it doesn't use any particular database. Instead, it only checks if the current user name and password are equal to some constants.

In real-world scenario, you will need to replace this condition with lookup of the user name with given password in your external database.

Put the following code inside the OnAuthentication method:

**[C#]**

```
// Check if the user was authenticated by the system
if (userInfo != null)
{
   return userInfo;
}

// Sample external user credentials
UserInfo usr = null;

// Not authenticated, authenticate from the external source
if ((username.ToLower() == "externaluser") && (password == "pass"))
{
   // Create base user record if user found
   usr = new UserInfo();
   usr.IsExternal = true;
   usr.UserName = "externaluser";
   usr.FullName = "external user";
   usr.Enabled = true;

   // Init user sites and roles if requested
   Hashtable rolesTable = new Hashtable();
   string siteName = CMSContext.CurrentSite.SiteName;
   // Assign user to the current site
   usr.SitesRoles[siteName.ToLower()] = rolesTable;
   // Add new role "external role" and assign it to the user
   rolesTable["external role"] = 0;
}

// Return the user info
return usr;
```

## 59.11.5 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.

## 59.11.6 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;
```

## 59.12 Customizing the administration interface and web application events

This chapter describes how to customize the administration interface and web site functionality in the way which is safe for future system updates and compatible with the import/export functionality.

### 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#]**

```
/// <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;
    }
```

# 59.13 Using API and CMS Controls outside CMS project

You can use Kentico CMS API and Kentico CMS Controls also outside the standard CMS web site 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 2005 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 apllication 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

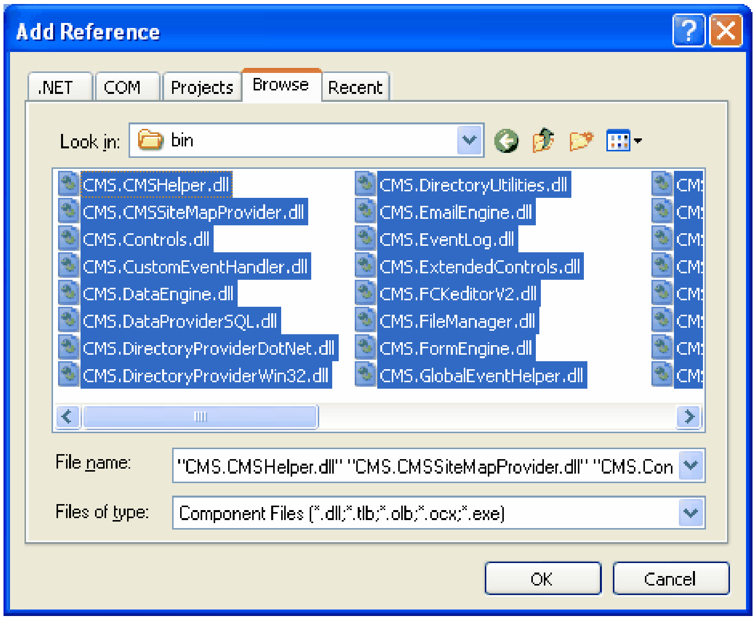
Add the connection string for the Kentico CMS database with name CMSConnectionString into the configuration/connectionStrings section of the web.config. The section will look like this:

```
<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

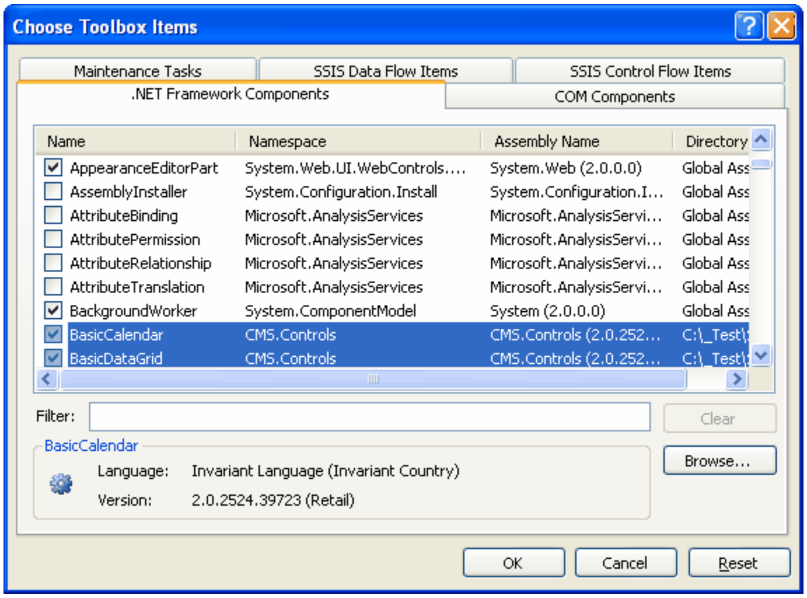
Now we will add a reference to Kentico CMS libraries. Right-click your web project in the **Solution Explorer** and and choose **Add reference...** Choose the **Browse** tab and locate the CMS web project's bin folder. Choose all libraries and click OK:



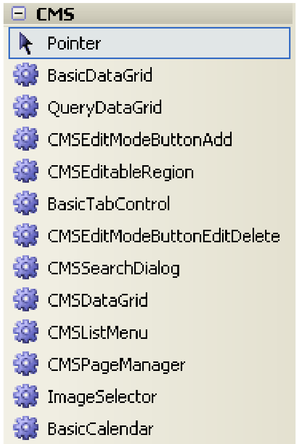
## Adding Kentico CMS Controls to your toolbox

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.

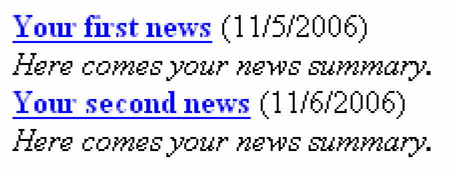
## Using Kentico CMS Controls to display content from Kentico CMS database

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 2005. 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 web site.

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()
```

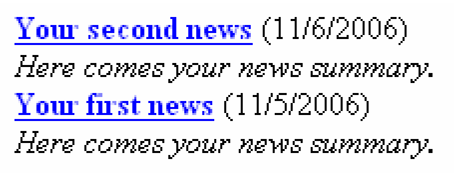
**Please note**

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 web site 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.

# Part XX

# 60 Import Utilities

## 60.1 Import overview

Kentico CMS provides two tools for importing content:

- File import module that enables import of bunch of files from disk to the content tree.
- SQL import utility that enables import of data from MSSQL database table as documents to the content tree.

## 60.2 SQL import

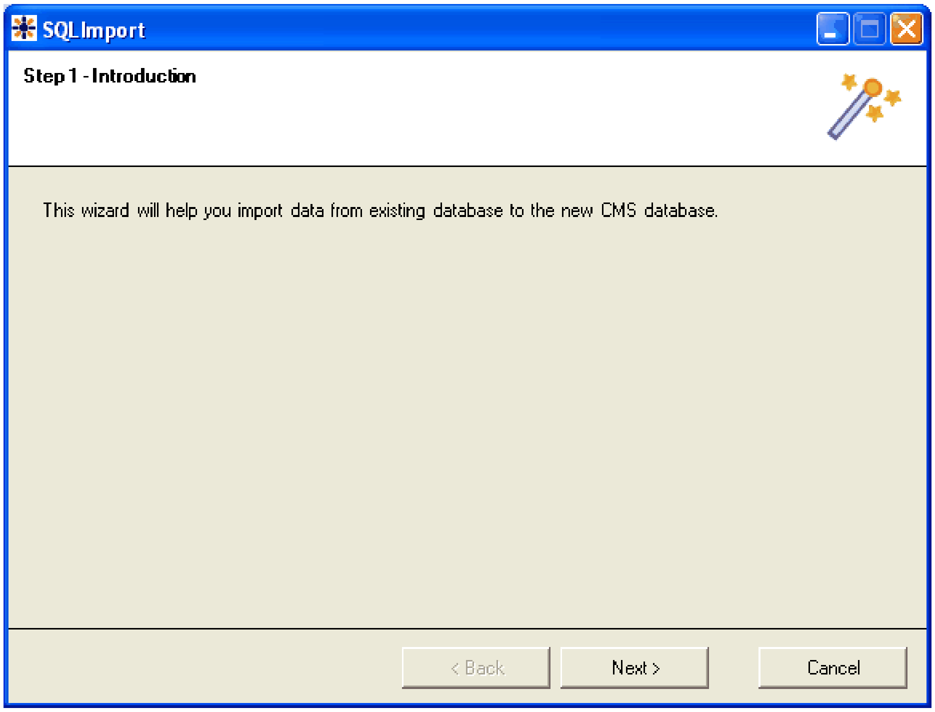
### 60.2.1 How to use the SQL import utility

The SQL Import utility allows you to import data from a Microsoft SQL Server database as new documents into Kentico CMS content repository. 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.

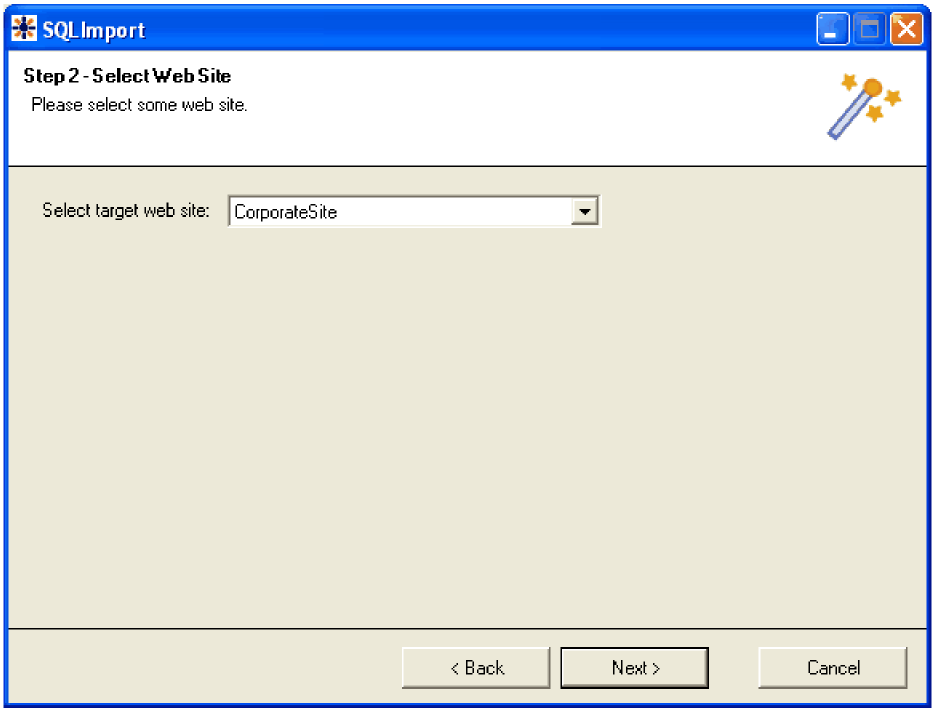
Before you run the SQL Import utility, you need to configure its parameters. The utility is stored in folder <Kentico CMS installation in Program Files>\SQLImport. Open the SQLImport.exe.config file and set the following values:

| 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 with files to be imported (in case you are importing files together with data). | <add key="ImportFilesDiskPath" 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" /> |

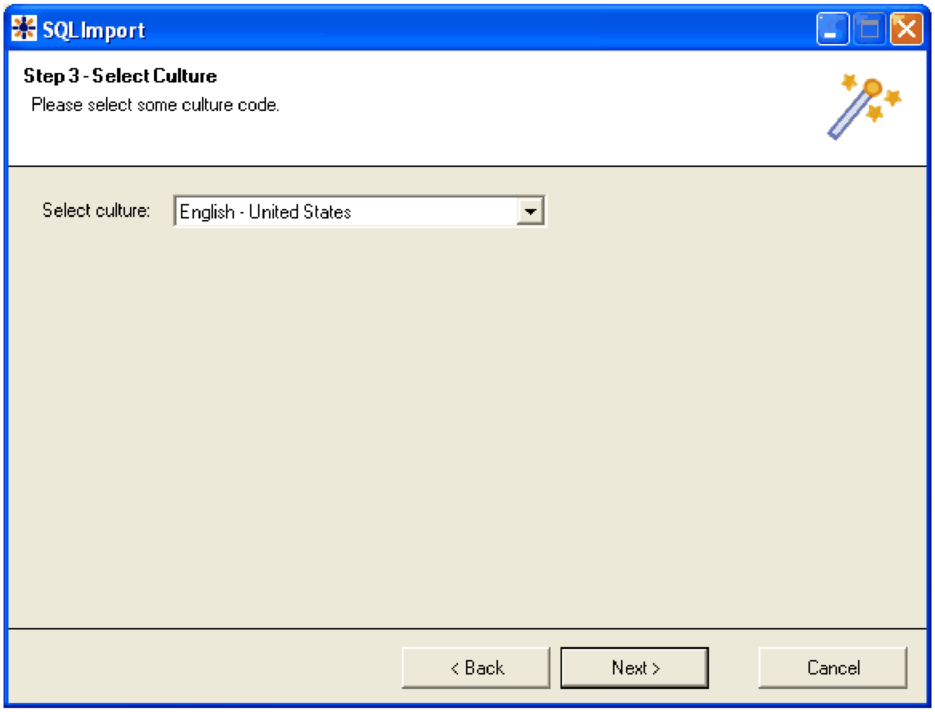
Now you can start the import utility using SQLImport.exe. The import wizard starts:



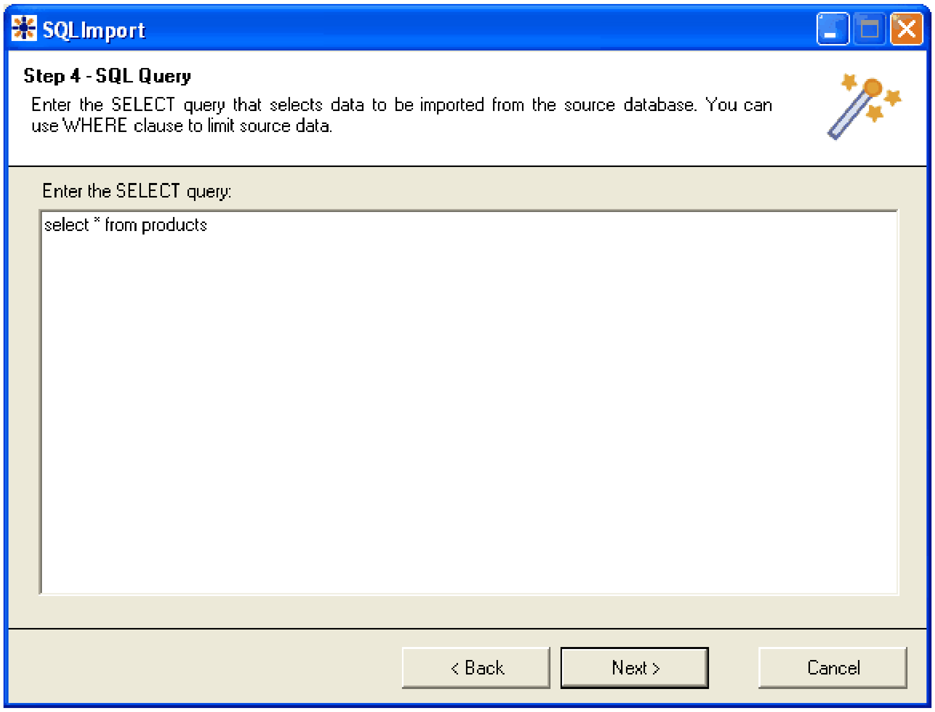
Click **Next**. Select the target web site:



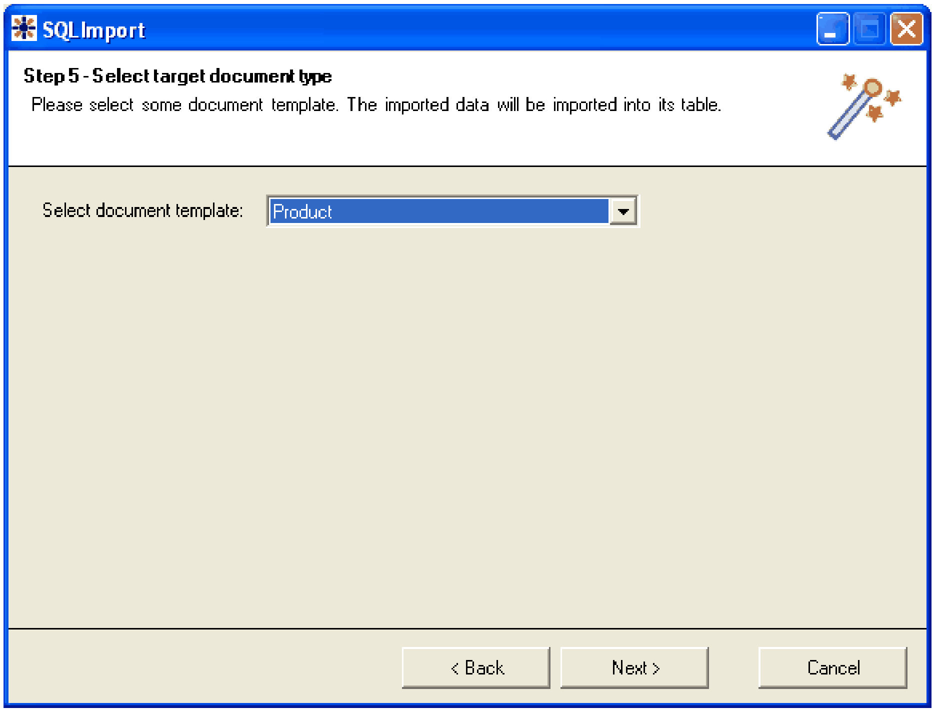
Click **Next**. Select the target content culture:



Click **Next**. Enter the SQL query that will return the data to be imported from the source database:

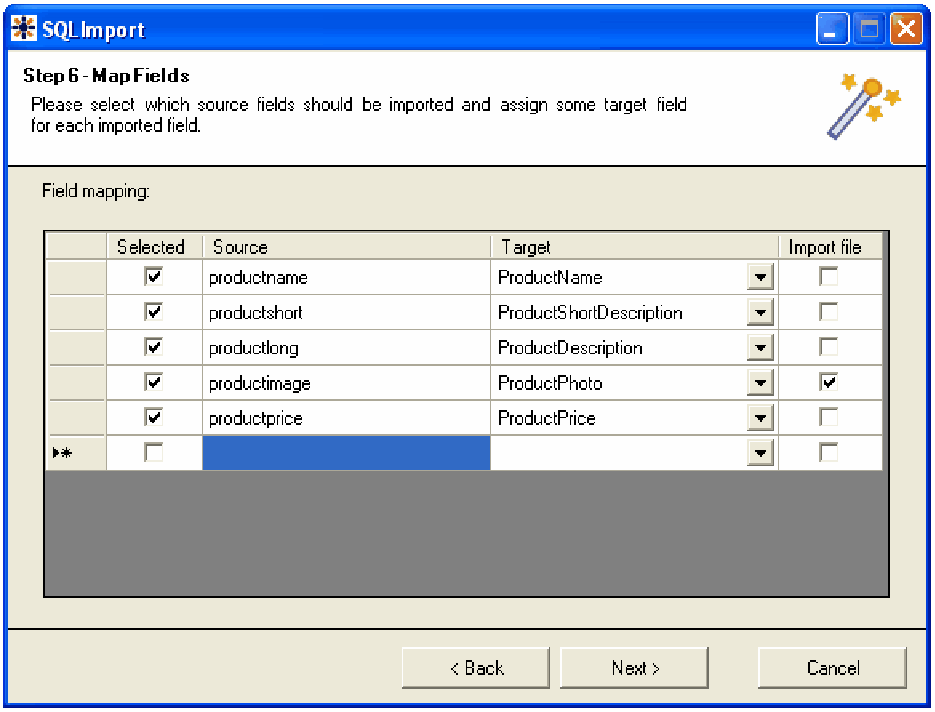


Click **Next**. Select the type of the imported documents.

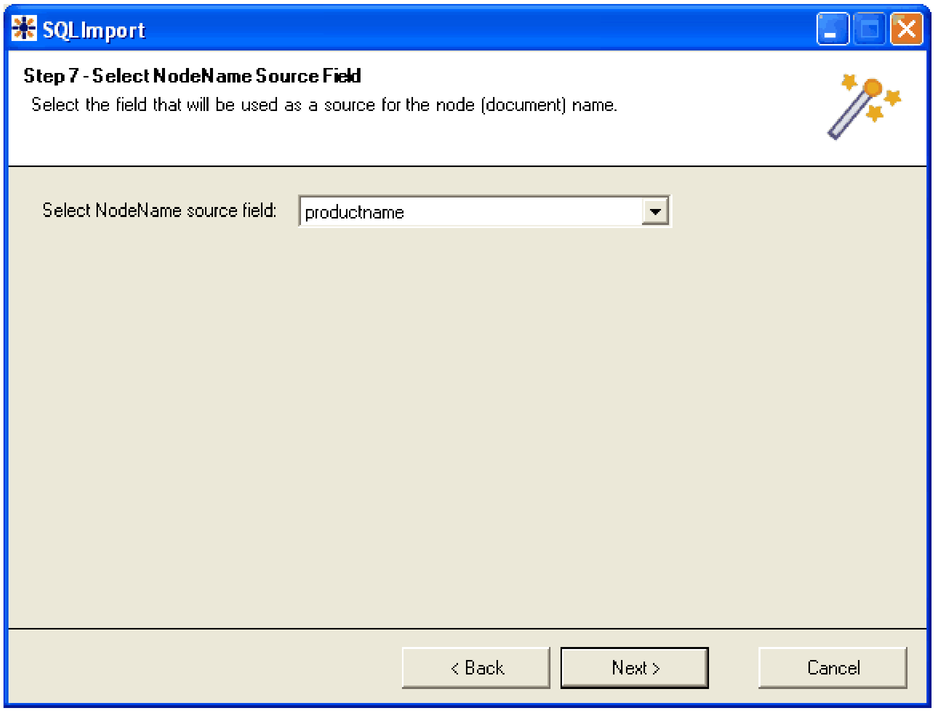


Click **Next**. Map the source and target fields and check the fields that should be imported. You can add the source fields manually multiple times if you need to import one source field into several target fields.

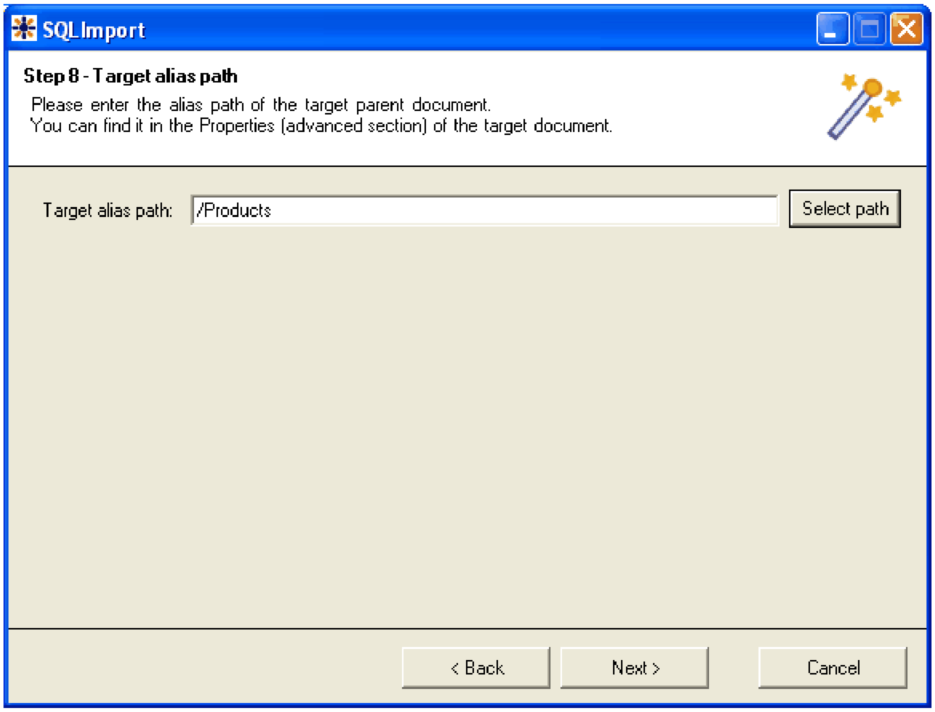
If the source field contains the path to the file, check the box **Import file**. In this case, the source field must contain a relative path to the path in the ImportFilesDiskPath web.config parameter - both values are joined and the file is imported from the specified location.



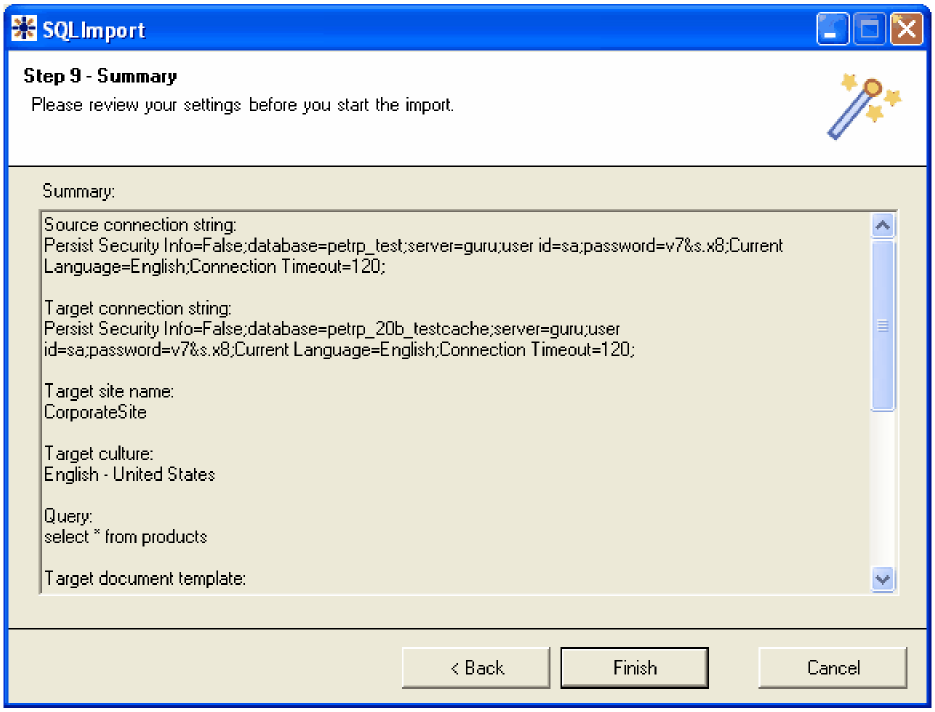
Click **Next**. Select the field that will be used for the name of the document:



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 Kentico CMS database.

# Part XX

# 61 Search engine

## 61.1 Overview

Kentico CMS search engine 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 an appropriate document type in Site Manager -> Development -> Document types -> ... select document type ... -> edit the searchtree query.

The common fields (such as document name) are searched using the cms.root.searchdocuments query.

The 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 chapter Configuratin of full-text search in files. 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 web site using the web parts in the Full-text search category (search dialog, 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 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 web site sections** from search by setting the **Exclude documents from 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 web site section can be excluded using e.g. **/news/%**. You can enter multiple values separated with a semicolon (;).

### 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

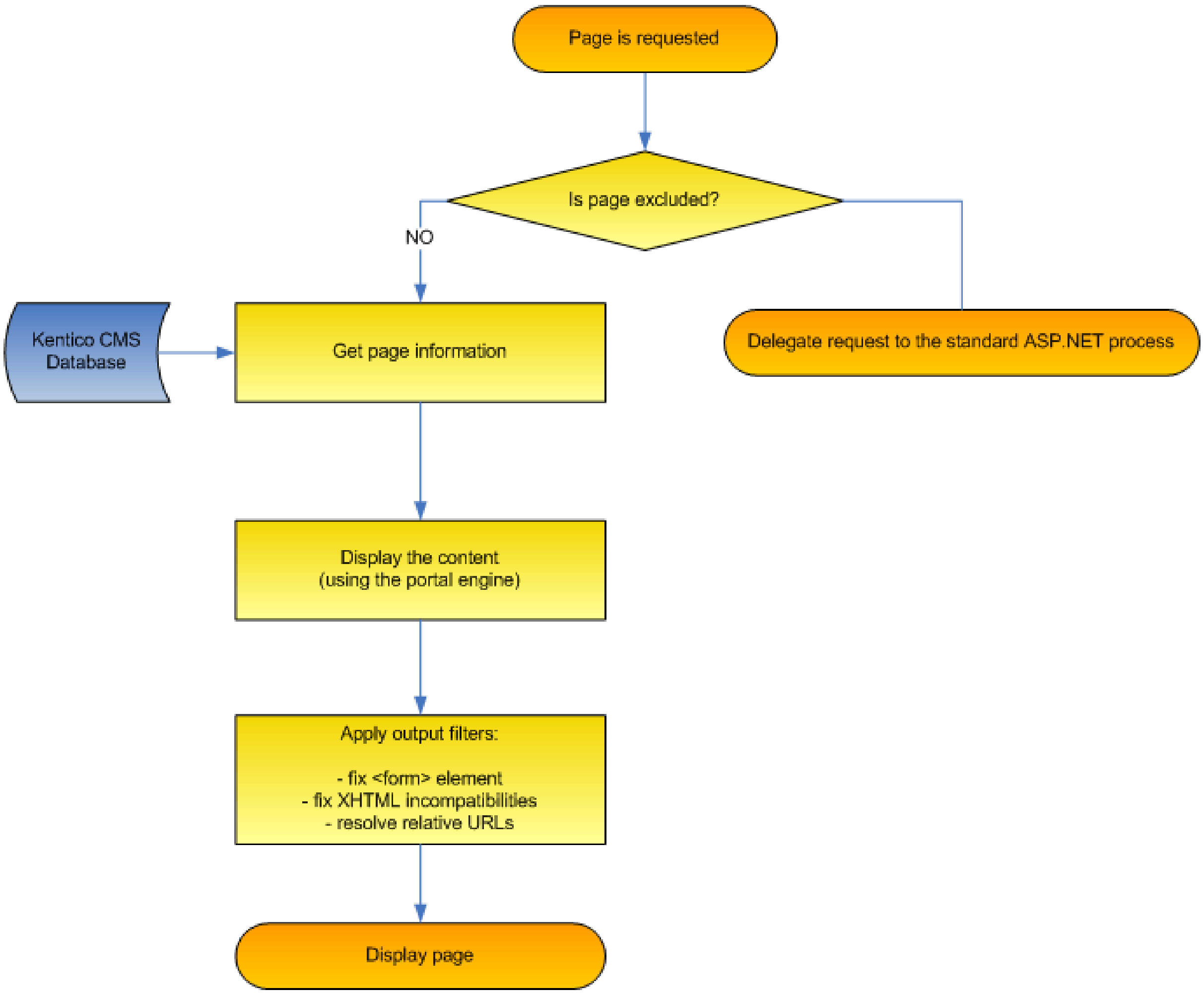
If you need to integrate a custom search engine or make additional modifications to the search results returned by the standard search engine, you can develop your own search provider. You can find more details in chapter Custom Search Provider.

# Part XX

# 62 Page processing and URLs

## 62.1 Overview

Kentico CMS processes URLs using a **URL rewriting engine**. This engine ensures displaying of the correct page based on the required friendly (smart)URL. After the page is processed by the URL rewriting engine and it also runs the output filters that ensure additional changes in the rendered HTML code. The following figure shows the page processing step-by-step:




## 62.2 URL rewriting

Kentico CMS uses a system of friendly URL addresses. It allows you to use URLs like:

**`http://www.mydomain.com/products/kentico-cms.aspx`**

instead of

**`http://www.mydomain.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.mydomain.com/products/kentico-cms.aspx`** and looks up the web site based on the domain name (either on the main domain name or domain aliases). If it doesn't find any running web site, it displays the page **`/cmsmessages/invalidwebsite.aspx`**. If the domain name with 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 with current culture. It can optionally search also for other 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 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 for its displaying and call the appropriate page template like **`/products.aspx?aliaspath=/prodcuts/kentico-cms`**. The page template is responsible for displaying of the document. If the document doesn't have any page template specified, the URL rewriting engine tries to find the page template of the parent and use it. If the page is managed by a 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 to search engines (SEO friendly).
- They show user where she is located on the web site.
- You can easily send the URL of the document to your friend and she will see the same page with particular document.

## 62.3 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:

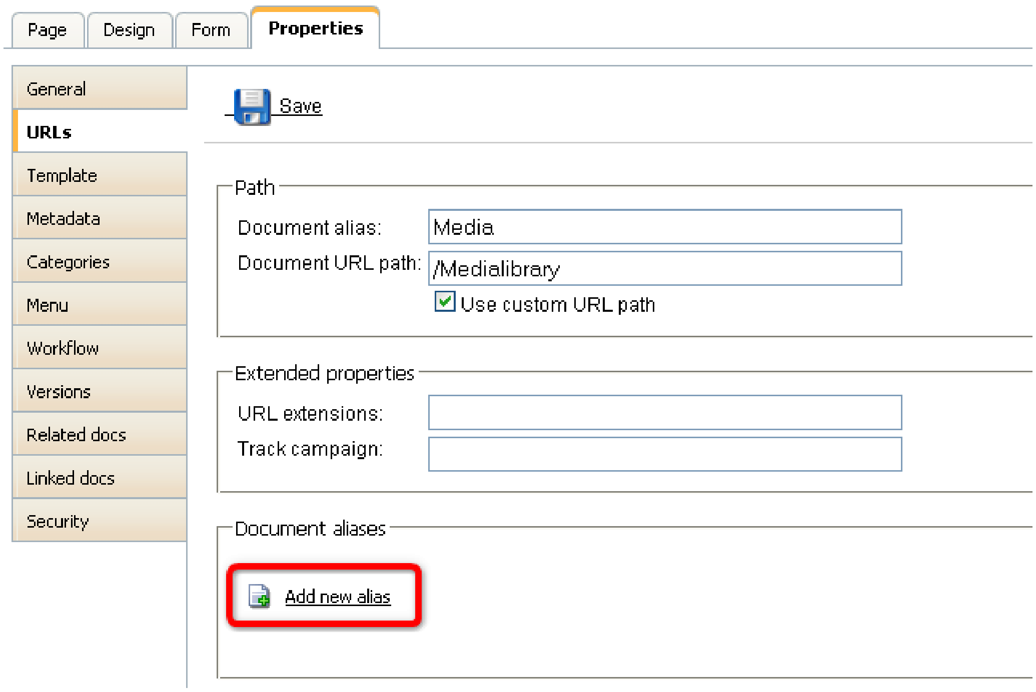
- **Document alias** - this is the unique name of the document in the given section of the web site; if **Media** is the value, then **<domain>/Media.aspx** is the URL under that the document can be accessed by default

- **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:

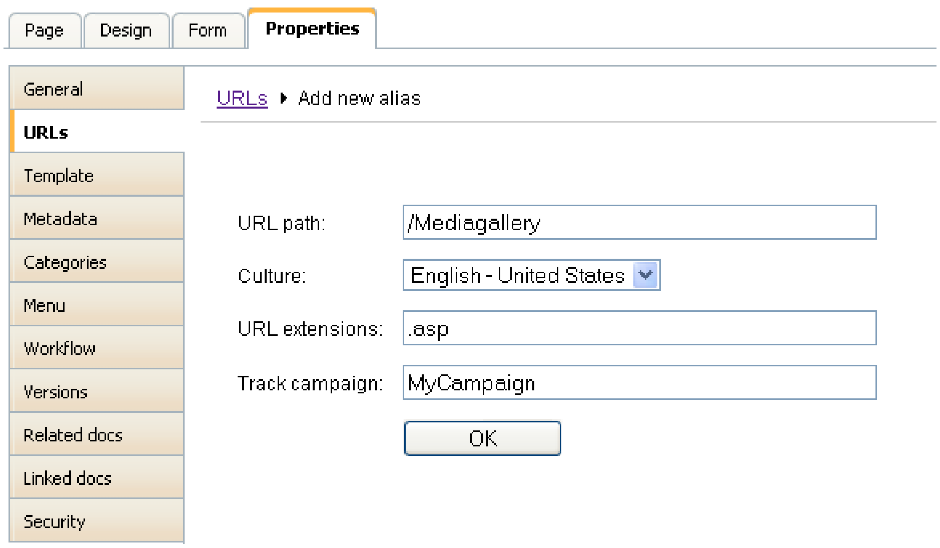
1) Sign in to CMS Desk, select some document from the content tree and switch to its **Properties -> URLs** tab.

2) Click the **Add new alias** link in the **Document aliases** section.



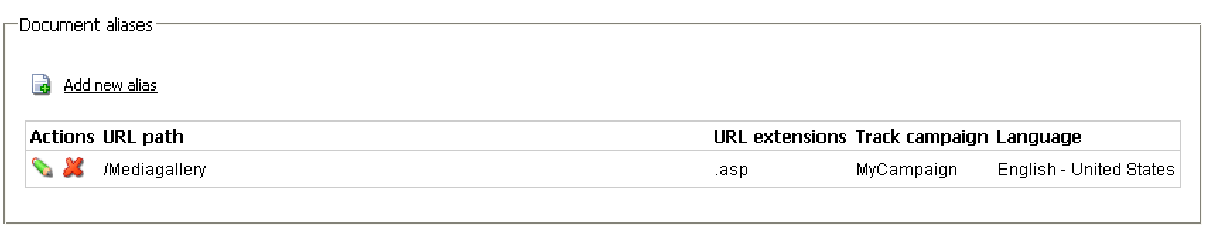
3) Enter the following details:

- **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.

# 62.4 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.mydomain.com/products/kentico-cms.aspx`**.

You can also use **custom extensions**, such as **`.htm`**, **`.html`** or any custom extension. However, in this case, you need to configure the system as described in Configuration of custom URL extensions.

Alternatively, you can use **URLs without extensions**, such as **`http://www.mydomain.com/products/kentico-cms`**. In this case, you need to configure the system as described in Configuration of extension-less URLs.

## Excluding URLs from the CMS engine

If you need to add your own pages to the web site, 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 -> Excluded URLs** value (you can enter several URLs separated with 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 a space.

You can add additional forbidden characters by entering them (without any separator) into the **Site Manager -> Settings -> URLs -> Forbidden URL Characters** value.

The forbidden characters are removed by default. You can specify the character that will replace forbidden characters in the **Site Manager -> Settings -> URLs -> Forbidden Characters Replacement** value.

## 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 -> Default Url Path Prefix** value.
The URLs will then look like this:
**`http://www.mydomain.com/myprefix/products/kentico-cms.aspx`**.

## Automatic creation of new document aliases

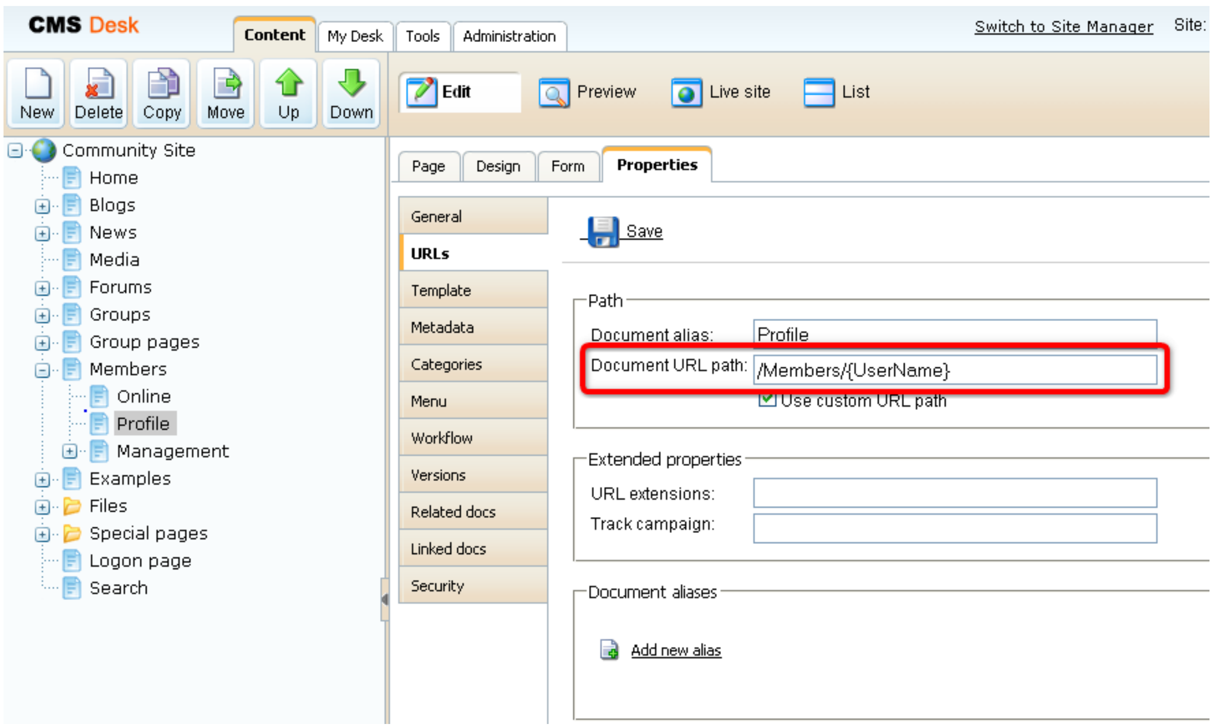
If you check the **Site Manager -> Settings -> URLs -> Remeber original URLs when moving documents** check-box, new document aliases will be automatically created when a new extension or URL path is set.

# 62.5 Wildcard URLs

## Wildcard URLs

Wildcard URLs are one of the new features in Kentico CMS 4.0. You can find an example of how this works on the sample Community starter site. The **Members -> Profile** page uses wildcard URLs to display users' 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.

# 62.6 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.mydomain.com/getdoc/016fad52-0d69-46d5-80dc-daec9173c0c7/
Products.aspx
```

It's an equivalent of

**http://www.mydomain.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 in CMS Desk -> Content -> Properties -> General, in field Node GUID.

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.mydomain.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.mydomain.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 to some document, such as DOC, PDF or image uploaded in the product form, you need to use a URL in the following format:

http://www.mydomain.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 document GUID. It's a GUID of the file in the CMS_Attachment table. You can find this GUID if you display the image in the CMS Desk and see its URL.

The **`<file name>`** value can contain any text.

You can find more details on available parameters in the following chapter GetFile.aspx parameters

## 62.7 GetFile.aspx parameters

The GetFile.aspx script is used in many cases to retrieve the uploaded from from the database. It is called whether you use /getdoc, /getattachment or direct URL based on alias path of the 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 **guid** parameter. | |
| width | Resizes the image to specified width (in pixels). | 100 |
| height | Resizes the image to 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<br><br>or<br><br>attachment |

## 62.8 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 them to **Site Manager -> Settings -> URLs -> Excluded Output Form Filter URLs** value, separated with 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 them to **Site Manager -> Settings -> URLs -> Excluded XHTML Filter URLs** value, separated with a semicolon (;).

The XHTML errors may also be fixed in the WYSIWYG editor when they are saved. This can be configured globally in the CMSWYSIWYGFixXHTML web.config parameter (supported values are "true" and "false").

### Resolve filter

The resolve filter changes relative URL in format **`~/mypage1/mypage2.aspx`** to **`/application/mypage1/mypage2.aspx`** (application running in a subfolder) 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 them to **Site Manager -> Settings -> URLs -> Excluded Resolve Filter URLs** value, separated with a semicolon (;).

# Part XX

# 63 System tables and custom fields

## 63.1 System tables

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's 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's highly recommended that you always allow empty values in the newly created field, so that the system procedures and existing dialogs keep working.

## 63.2 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";
```

# Part XX

# 64 Site settings and web.config parameters

## 64.1 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 chapter Web.config parameters.

### Site settings

The site settings are stored in Kentico CMS database. You can configure them in **Site Manager -> Settings**. There are two types of settings:

- **Global**
- **Site-specific** - you can edit them after you choose the relevant web site in the drop-down list in the upper left corner.

The site-specific settings may inherit global values.

You can find more details on particular settings key when you mouse-over the key name - you will see a pop-up window that displays the key description. Detailed descriptions can also be found in Context help, which can be displayed by clicking the green "?" icon at the top right corner of the page.

## 64.2 System settings

System settings can be done in **Site Manager -> Settings -> System**.

## 64.3 Web.config parameters

The system settings include appSettings keys and other settings, such as connection string placed in appropriate sections of the web.config file.

### appSettings keys

appSettings keys are stored in section **/configuration/appSettings**.

| 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:<br><br>• **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. This is the default value. | <add key="CMSDirectoryProviderAssembly" value="CMS.DirectoryProviderDotNet" /><br><br>or<br><br><add key="CMSDirectoryProviderAssembly" value="CMS.DirectoryProvid |

| | | |
|---|---|---|
| | • **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. | erWin32" /> |
| CMSProgrammingLanguage | Indicates the programming language used in transformations and in custom code added to web parts.<br><br>The default value is C#. | <add key="CMSProgrammingLanguage" value="C#" /><br><br>or<br><br><add key="CMSProgrammingLanguage" value="VB" /> |
| CMSSearchProviderAssembly | Name of the assembly that is used for full-text search. The default assembly is CMS.SearchProviderSQL. | <add key="CMSSearchProviderAssembly" value="CMS.SearchProviderSQL" /> |
| 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. | <add key="CMSStagingAcceptAllCertificates" value="true" /> |
| CMSTrialKey | Contains a temporary trial key. You can remove this value after installation. | |
| CMSUseAutomaticScheduler | Indicates if automatic scheduling should be used. The default value is false.<br><br>Automatic scheduler periodically requests the cmspages/scheduler.aspx page which ensures that scheduled tasks are processed automatically.<br><br>If it's turned off (false - by default), the tasks are processed at the end of page request. If it's true, the cmspages/scheduler.aspx page is requested periodically. | <add key=" CMSUseAutomaticScheduler" value="false" /> |
| CMSUseCustomHandlers | Indicates if custom handlers should be executed to process the system events.<br><br>The default value is false. | <add key="CMSUseCustomHandlers" value="true" /> |
| CMSWebFarmServerName | Code name of the web farm server. This value is used for native web farm synchronization support. | <add key="CMSWebFarmServerName" value="server1" /> |

| | | |
|---|---|---|
| CMSWYSIWYGFixXHTML | Indicates if the WYSIWYG editor should automatically try to fix XHTML incompatibilities in the code.<br><br>Supported values are "true" and "false". The default value is true. | <add key=" CMSWYSIWYGFixXHTML" value="true" /> |
| CMSUseIFrameForHTMLEditorToolbar | Indicates if the WYSIWYG editor shared toolbar should be displayed in an iframe to avoid issues with design in the editing mode. The default value is false. | <add key="CMSUseIFrameForHTMLEditorToolbar" value="true" /> |
| CMSUseVirtualPathProvider | Indicates if the virtual path provider should be used (true by default). Before you switch off the virtual path provider, please read chapter Pre-compilation (Publish function). | <add key=" CMSUseVirtualPathProvider" value="false" /> |
| 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. | <add key="CMSDefaultSpellCheckerCulture" value="en-US" /> |
| CMSShowLogonCultureSelector | Indicates if the logon page should display the CMS Desk should display a dropdownlist with available user interface languages. | <add key=" CMSShowLogonCultureSelector" value="false" /> |
| CMSUsePermanentRedirect | Indicates if Response.Redirect (302, default mode) should be replaced with static redirect (301). | <add key="CMSUsePermanentRedirect" value="true" /> |
| CMSShowWebPartCodeTab | Indicates if the Code tab is displayed in the web part properties. This parameter is used for backward compatibility. | <add key=" CMSShowWebPartCodeTab" value="true" /> |
| CMSShowWebPartBindingTab | Indicates if the Binding tab is displayed in the web part properties. This parameter is used for backward compatibility. | <add key="CMSShowWebPartBindingTab" value="true"/> |
| CMSShoppingCartExpirationPeriod | Number of days after which the 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. | <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. | <add key="CMSUseCurrentSKUData" value="true"/> |
| CMSFullClientCache | Indicates if full client cache is enabled. | <add key=" CMSFullClientCache" value="true"/> |
| CMSRenderGeneratorName | Indicates if generator meta tag is | <add key=" |

| | | |
|---|---|---|
| | rendered. | CMSRenderGeneratorName" value="true"/> |
| CMSClearFieldEditor | Indicates if new field is empty (true) or has attributes from another field (false). | <add key=" CMSClearFieldEditor" value="true"/> |
| CMSDeskKeepAlive | Indicates if refresh meta tag is inserted (true) or not (false). | <add key=" CMSDeskKeepAlive" value="true"/> |
| CMSShowTemplateASPXTab | Indicates if ASPX tab is shown. | <add key=" CMSShowTemplateASPXTab" value="true"/> |
| CMSDatabaseCulture | Specifies the default database culture. | <add key=" CMSDatabaseCulture" value="en-us"/> |
| CMSDataProviderAssembly | Specifies custom data provider assembly. | <add key=" CMSDataProviderAssembly" value=" CMS.CustomDataProvider "/> |
| CMSWebFarmEnabled | Indicates if Web farms are enabled or not. | <add key=" CMSWebFarmEnabled" value="true"/> |
| CMSDisposeConnectionAfterClose | If true, disposes connection after the connection is closed. | <add key=" CMSDisposeConnectionAfterClose" value="true"/> |
| CMSUseSessionCookies | Indicates if session cookies are used or not. | <add key=" CMSUseSessionCookies" value="true"/> |
| CMSImportWindowsRoles | Indicates if windows roles are imported or not. | <add key=" CMSImportWindowsRoles" value="true"/> |
| CMSDefaultUICulture | Specifies the default UI culture. | <add key=" CMSDefaultUICulture" value="en-us"/> |
| CMSFileScriptTimeout | Sets script timeout in seconds. | <add key=" CMSFileScriptTimeout" value="7200"/> |
| CMSUseExtensionOnPostback | Indicates if extension is used on postback. | <add key=" CMSUseExtensionOnPostback" value="true"/> |
| CMSUseSQLResourceManager | If true, SQL Resource manager is used to retrieve strings. | <add key=" CMSUseSQLResourceManager" value="true"/> |
| CMSLoadHTMLEditorByIFrame | Indicates if HTML editor is loaded by iframe. | <add key=" CMSLoadHTMLEditorByIFrame" value="true"/> |
| CMSUseFrameForHTMLEditorToolbar | Indicates if frame is used for editor toolbar. | <add key=" CMSUseFrameForHTMLEdit |

| | | |
|---|---|---|
| | | orToolbar" value="true"/> |
| CMSWYSIWYGFormatURL | If true, the editor runs the URL formatting. | <add key=" CMSWYSIWYGFormatURL" value="true"/> |
| CMSAllowCheckIOPermissions | Indicates if IO (write) permissions checking is allowed. | <add key=" CMSAllowCheckIOPermissions" value="true"/> |
| CMSEmailTransferEncoding | Specifies e-mail transfer encoding. | <add key=" CMSEmailTransferEncoding" value=" TransferEncoding.Unknown (-1)"/> |
| CMSEmailProviderAssembly | Specifies custom email provider assembly. | <add key=" CMSEmailProviderAssembly" value=" CMS.CustomEmailProvider "/> |
| CMSCheckParameters | Indicates if parameters checking is allowed. | <add key=" CMSCheckParameters" value="true"/> |
| CMSReportCheckParameters | If true, exception reporting the parameters is thrown when the parameters do not match. | <add key=" CMSReportCheckParameters" value="true"/> |
| CMSLogEvents | Indicates if logging is enabled. | <add key="CMSLogEvents" value="true"/> |
| CMSLogMetadata | Indicates if logging of metadata events is enabled. | <add key=" CMSLogMetadata" value="true"/> |
| CMSLogKeepPercent | Coefficient for log deletion, keeps the specified percentage of log items alive and deletes the log by batch when the set percentage are exceeded. | <add key=" CMSLogKeepPercent" value="10"/> |
| CMSCustomHandlersAssembly | Specifies custom event handler assembly. | <add key=" CMSCustomHandlersAssembly" value=" CMS.CustomEventHandler "/> |
| CMSWorkflowSendEmailToModerator | If true, e-mails will be sent to the current moderator. | <add key=" CMSWorkflowSendEmailToModerator" value="true"/> |
| CMSControlElement | Overrides the generation of the SPAN tag with custom tag. | <add key=" CMSControlElement" value="div"/> |
| CMSUseParsedSelfClose | Indicates if parsed self close tags operation (faster) are used instead of standard self close filter. | <add key=" CMSUseParsedSelfClose" value="true"/> |
| CMSEnableOrderItemEditing | If true, order item's parameters, such as order item name and order item unit | <add key=" CMSEnableOrderItemEditing |

| | | |
|---|---|---|
| | price, can be modified when editing an existing order. | " value="true"/> |
| CMSUseMetaFileForProductImage | Indicates if meta files should be used for product images. | <add key=" CMSUseMetaFileForProductImage" value="true"/> |
| CMSUseCustomEcommerceProviders | Specifies whether to use custom e-commerce provider handlers. | <add key=" CMSUseCustomEcommerceProviders" value="true"/> |
| CMSCustomEcommerceProviderAssembly | Specifies the custom e-commerce provider assembly. | <add key=" CMSCustomEcommerceProviderAssembly" value=" CMS.CustomECommerceProvider"/> |
| CMSVirtualPathProviderAssembly | Specifies the Custom virtual path provider assembly. | <add key=" CMSVirtualPathProviderAssembly" value=" CMS.CustomVirtualPathProvider"/> |
| CMSGetFileEndRequest | If true, ApplicationInstance.CompleteRequest() is used instead of Response.End() in CompleteRequest method. | <add key=" CMSGetFileEndRequest" value="true"/> |
| CMSLogEventsToFile | If true, events are also logged into file. | <add key=" CMSLogEventsToFile" value="true"/> |
| FCKeditor:BasePath | Specifies FCK editor base path. | <add key=" FCKeditor:BasePath" value="/FCKeditor/"/> |
| FCKEditor:UserFilesPath | Specifies path to user files. | <add key=" FCKEditor:UserFilesPath" value="/UserFiles/"/> |
| CMSDefaultUserID | Specifies default user ID. | <add key=" CMSDefaultUserID" value="53"/> |
| ImportFilesDiskPath | Specifies path to files which may be imported. | <add key=" ImportFilesDiskPath" value="C:\Temp"/> |
| UICulture | Specifies user interface culture. | <add key="UICulture" value="en-us"/> |

## 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 |
|---|---|---|
| CMSTransactionIsolationLev | Isolation level for explicit transactions. | <add |

| | | |
|---|---|---|
| el | | key="CMSTransactionIsolationLevel" value="ReadCommitted" /> |
| CMSDefaultIsolationLevel | Isolation level for all queries, even without transactions. It's used only if the value is not ReadCommitted. The settings below modify the behavior for ReadUncommitted. | <add key="CMSDefaultIsolationLevel" value="ReadUncommitted" /> |
| CMSUseDefaultIsolationLevelOnlyWithOpenTransactions | If true, the default isolation level is used only when some transaction is already running. | <add key="CMSUseDefaultIsolationLevelOnlyWithOpenTransactions" value="true" /> |
| CMSAllowSimultaneousTransactions | If false, there can be only one transaction running at the same time. | <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. | <add key="CMSMaxTransactionWaitTimeout" value="1" /> |

## Special settings for file synchronization on web farms

By adding the following keys to your web.config, you can enable or disable synchronization of certain kind of files stored in the file system.

| Key | Description | Sample Value |
|---|---|---|
| CMSWebFarmSynchronizeFiles | General key determining if files should be synchronized (true) or not (false). | <add key="CMSWebFarmSynchronizeFiles" value="true" /> |
| CMSWebFarmSynchronizeAttachments | Enables/disables synchronization of atachments. | <add key="CMSWebFarmSynchronizeAttachments" value="true" /> |
| CMSWebFarmSynchronizeMetaFiles | Enables/disables synchronization of meta files. | <add key="CMSWebFarmSynchronizeMetaFiles" value="true" /> |
| CMSWebFarmSynchronizeMediaFiles | Enables/disables synchronization of media files. | <add key="CMSWebFarmSynchronizeMediaFiles" value="true" /> |
| CMSWebFarmSynchronizeBizFormFiles | Enables/disables synchronization of BizForm files. | <add key="CMSWebFarmSynchronizeBizFormFiles" value="true" /> |

| | | |
|---|---|---|
| CMSWebFarmSynchronizeAvatars | Enables/disables synchronization of Avatars. | <add key="CMSWebFarmSynchronizeAvatars" value="true" /> |
| CMSWebFarmSynchronizeForumAttachments | Enables/disables synchronization of forum attachments. | <add key="CMSWebFarmSynchronizeForumAttachments" value="true" /> |
| CMSWebFarmSynchronizeDeleteFiles | Enables/disables synchronization of deleted files. | <add key="CMSWebFarmSynchronizeDeleteFiles" value="true" /> |

**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;" />
```

# Part XXX

# 65 E-mail templates

## 65.1 E-mail templates

Kentico CMS sends e-mails for various purposes, such as workflow notification. You can modify the e-mail templates sent by the system so that they match your design and/or language.

You can modify e-mail templates in Site Manager -> Development -> E-mail templates.

There are two types of e-mail templates: global and site-specific. If there's no site-specific template, the global template is used instead.

The e-mail template has the following properties:

| | |
|---|---|
| Display name | The name of the template displayed to the users. |
| Code name | The name of the template used in the 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 copy recipients. These will get a copy of the e-mail, but won't see other recipient's addresses in the mail. |
| Subject | Subject of the e-mail. |
| Text | HTML code of the e-mail template. |
| Plain text | Plain text of the e-mail template. |

Example of the template text:

```
<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>
```

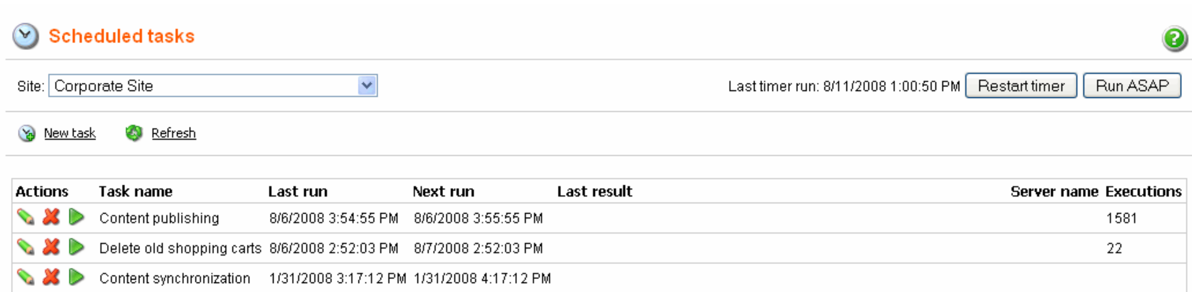
Please note that the template may contain special macro expressions that are resolved (merged) when the e-mail is sent, such as **{%approvedby%}**.

# Part XX

# 66 Scheduler

## 66.1 Scheduler overview

The Scheduler allows you to specify when specified scheduled task will be run. This is useful when you want some tasks to be run automatically in a specific time or time period. You can configure the scheduled tasks in **Site Manager -> Administration -> Scheduled tasks**.




The **Site** drop-down list is used for selecting a site. After selecting a site, a list of tasks scheduled for the site will be displayed in the table below.

Next to the site selection drop-down list, you can find the following two buttons:
- **Restart timer** - restarts the internal timer
- **Run ASAP** - immediately executes all tasks that are ready to be executed

The **New task** link redirects you to a page where new tasks can be scheduled. The **Refresh** link refreshes the tasks' list below.
Using the three icons next to each of the tasks, you can **Edit**, **Delete** or immediately **Execute** any of the listed tasks.

**Scheduling reliability**

Since the scheduling process runs within the ASP.NET process, it may not be executed if your web site is not running. This happens when the process is recycled without being started again (after a long period of web site inactivity). It means that your scheduled tasks will not be executed in such cases.

If you want to run the scheduling reliably, it's necessary to ensure that your web site is always running. You can do that by using some utility or an external service that requests the home page of your web site on a regular basis.

## 66.2 Scheduling a custom code

The process of scheduling includes two steps: writing code to perform required action and creating scheduled task.

### Writing task code

Task code must be placed into a specific class method.

1. Create a new library (assembly) as a part of your solution and a new class inside this library. The following example uses assembly name **CMS.Ecommerce** and class **CMS.Ecommerce.ShoppingCartCleaner**, however, you will need to use your own names. References to these dlls must be added to the newly created project:

```
CMS.Scheduler.dll
CMS.Staging.dll
CMS.DataEngine.dll
CMS.SettingsProvider.dll
```

2. Define method **Execute(TaskInfo task)** which will be performed when task is executed:

**[C#]**

```
using CMS.Scheduler;
namespace CMS.Ecommerce
{
    /// <summary>
    /// Provides an ITask interface to delete old shopping carts.
    /// </summary>
    public class ShoppingCartCleaner : ITask
    {
        public string Execute(TaskInfo task)
        {
            try
            {

                // Here comes the task code

                return null;
            }
            catch (Exception ex)
            {
                return (ex.Message);
            }
        }
    }
}
```

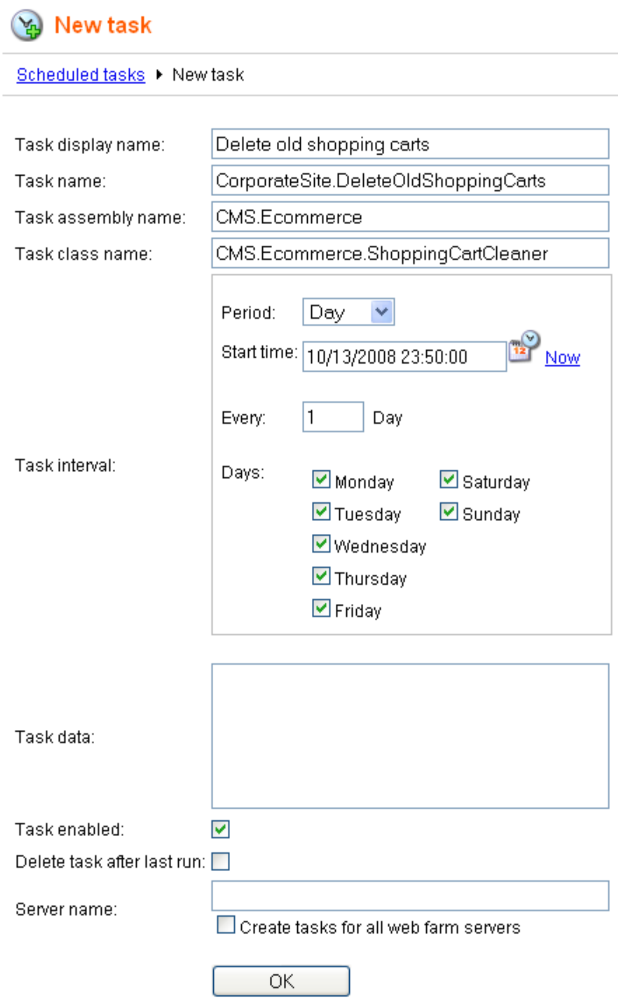
3. Compile the library.

## Creating a new task

**1.** Go to **Site Manager -> Administration -> Scheduled tasks.**
**2.** Select site the task should be scheduled for.
**3.** Click **New task**.
**4.** Specify task properties:

| | |
|---|---|
| Task display name | Name used for Scheduled Task. |
| Task name | Scheduled Task name used in code. |
| Task assembly name | Name of the assembly. |
| Task class name | Name of the class. |
| Task interval | Time interval when the task should be run. |
| Task data | Data which should be provided to assembly. |
| Task enabled | Indicates if the task is enabled. |
| Delete task after last run | Indicates if the task should be deleted after last run. |
| 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 scheduled.


New task
Scheduled tasks
New task
Task display name:
Delete old shopping carts
Task name:
CorporateSite.DeleteOldShoppingCarts
Task assembly name:
CMS.Ecommerce
Task class name:
CMS.Ecommerce.ShoppingCartCleaner
Period:
Day
Start time:
10/13/2008 23:50:00
Now
Every:
1
Day
Task interval:
Days:
Monday
Saturday
Tuesday
Sunday
Wednesday
Thursday
Friday
Task data:
Task enabled:
Delete task after last run:
Server name:
Create tasks for all web farm servers
OK

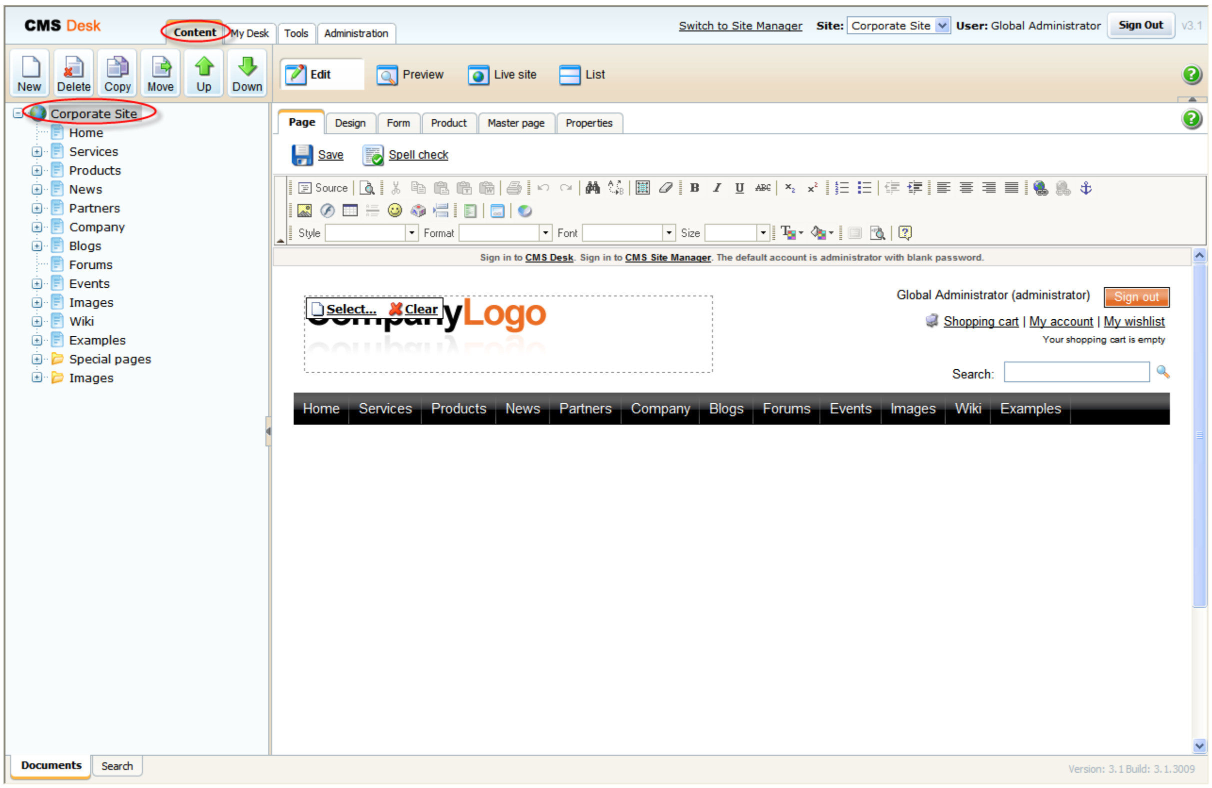
# Part XX

# 67 Rebranding

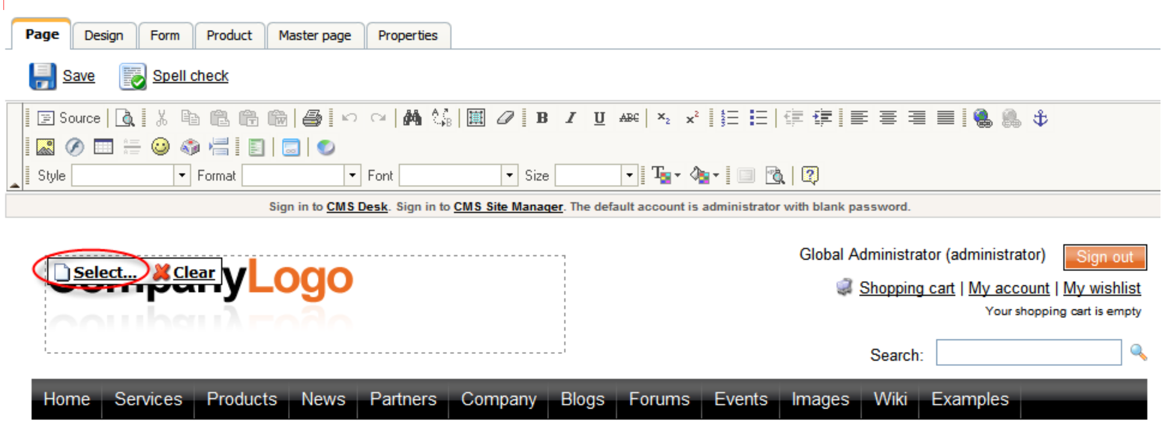
## 67.1 How to rebrand Kentico CMS with your name and design

### 67.1.1 Changing a logo in the header

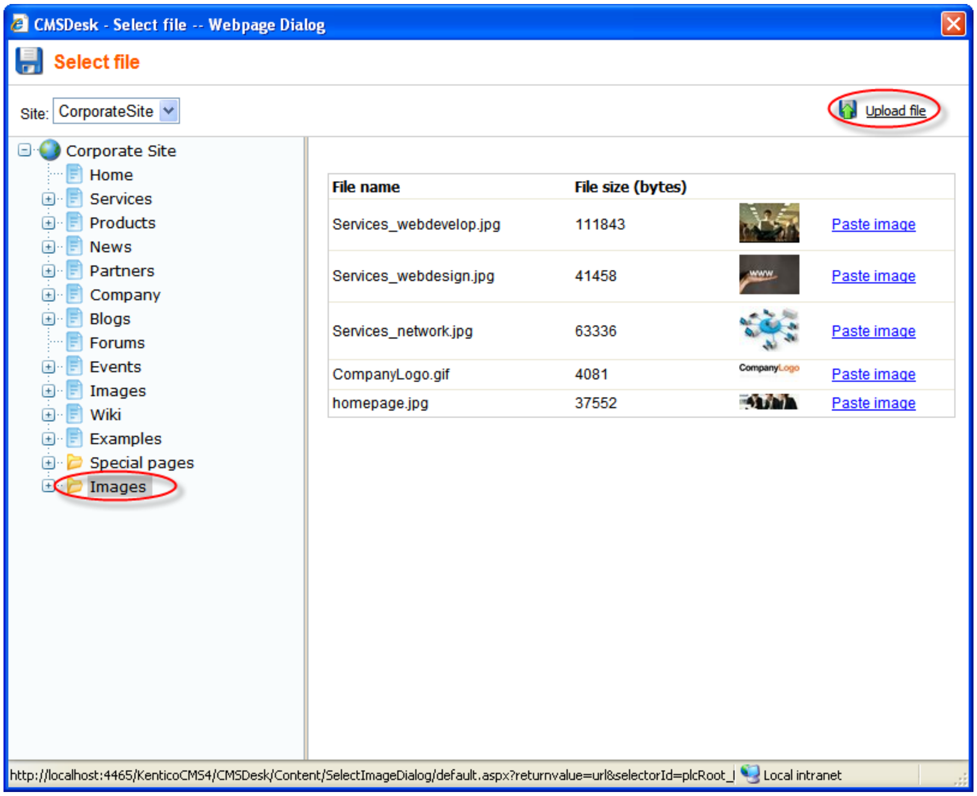
1. Go to **CMS Desk** -> **Content** -> **Corporate Site**.



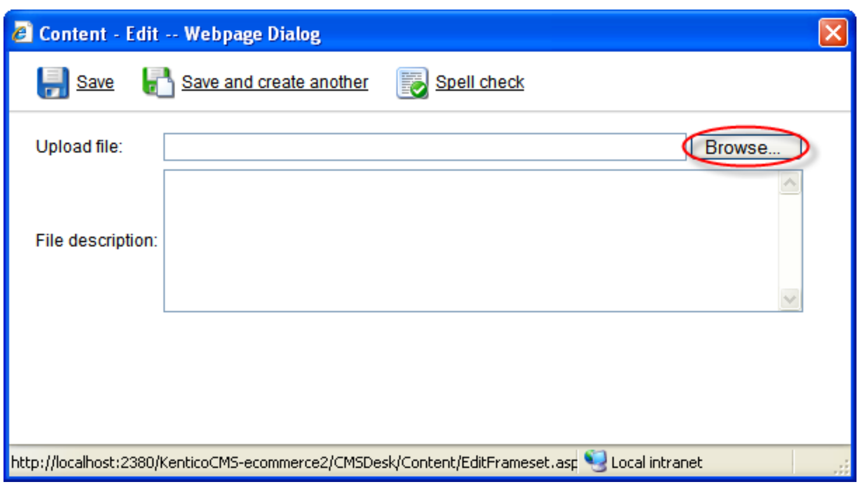

2. Click **Select ...** above the logo.



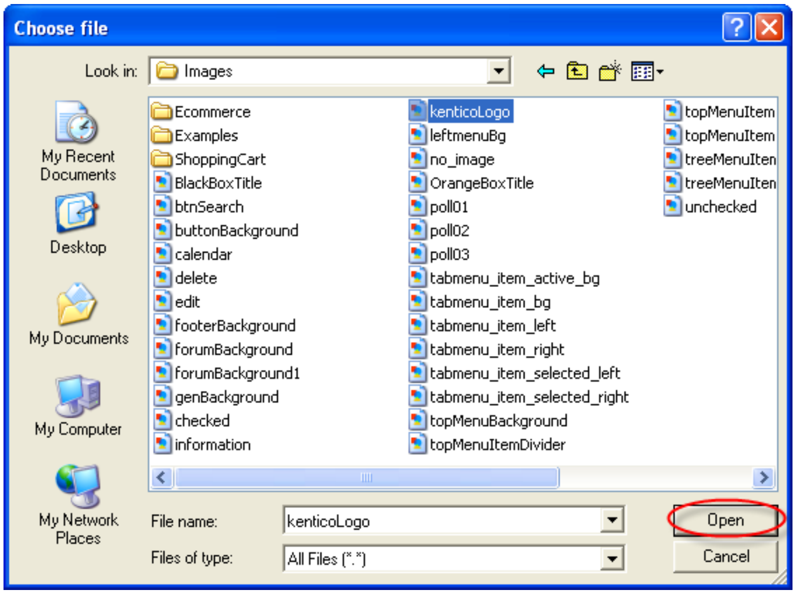
3. In the web page dialog, select **Images** in the content tree and click the **Upload file** button at the top-right.



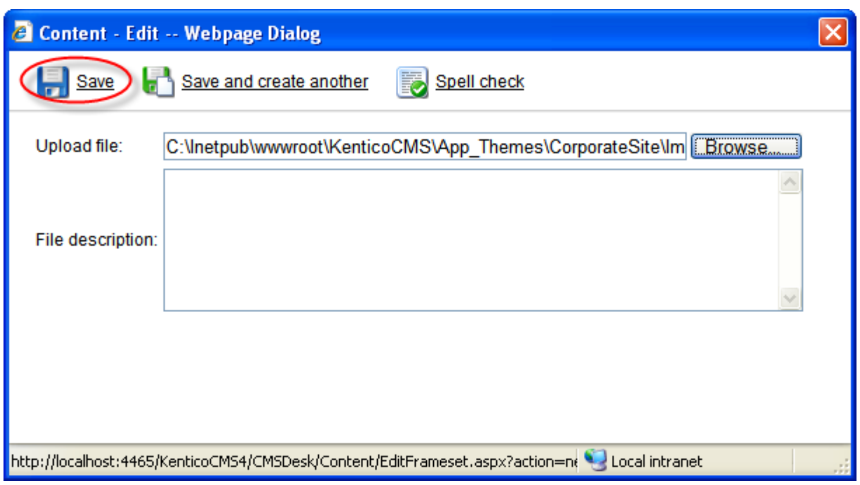

4. In the new dialog, click **Browse**.



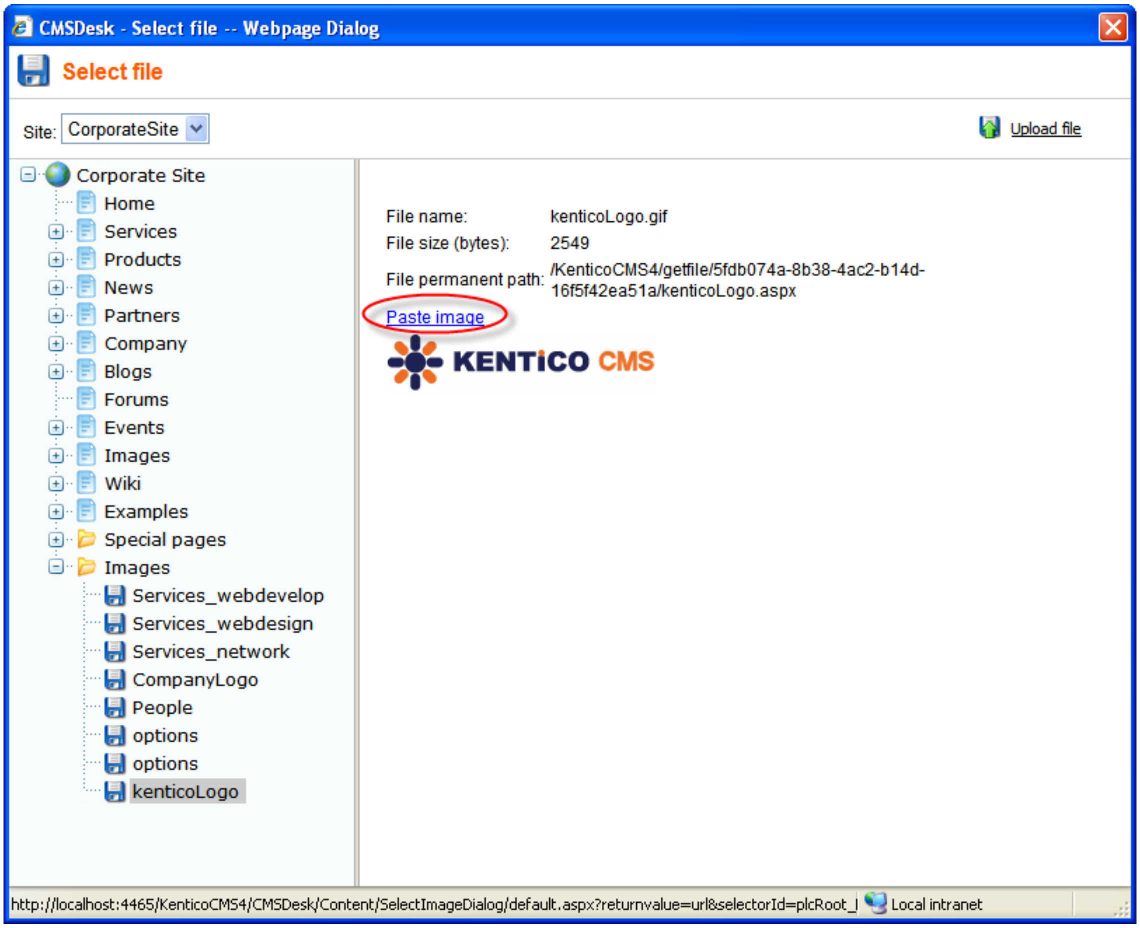
5. Find an image you want to upload and click **Open**.



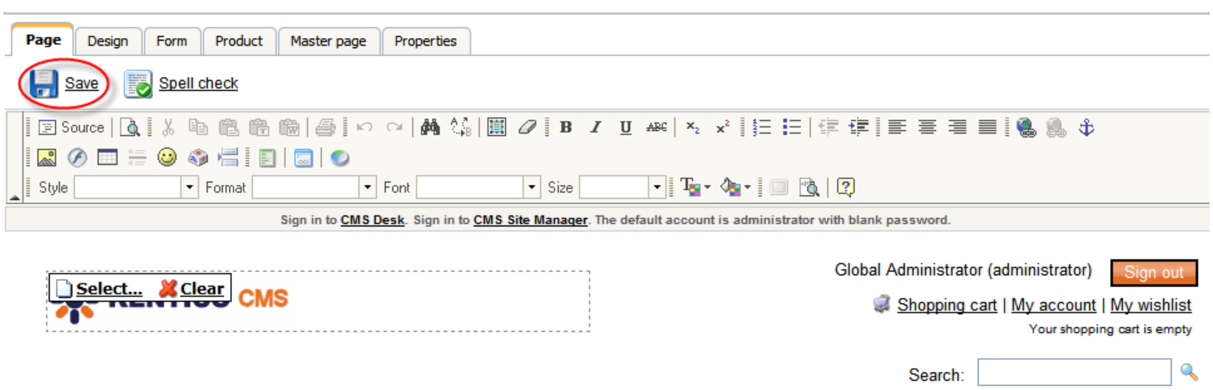

6. Click **Save**. Your new logo image is now ready to be posted on your web site.



7. Click **Paste image** right above the picture.



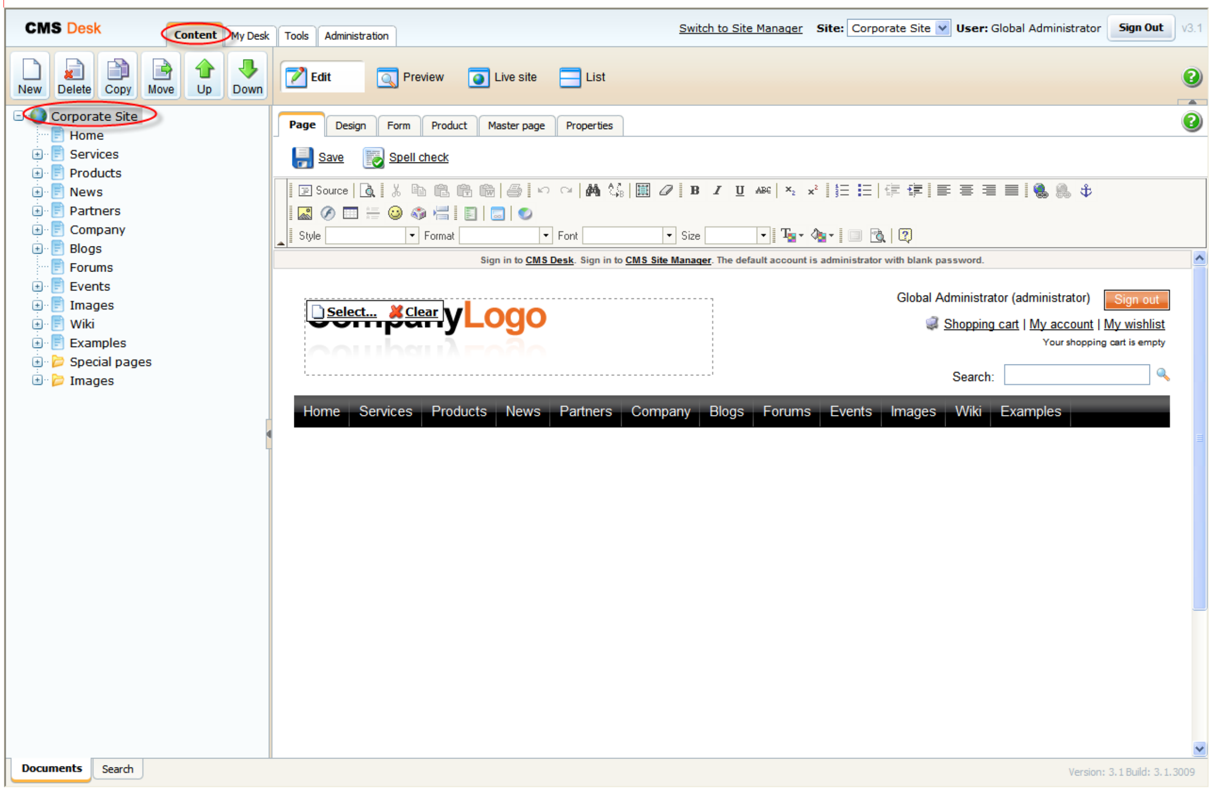

8. Now click **Save**.



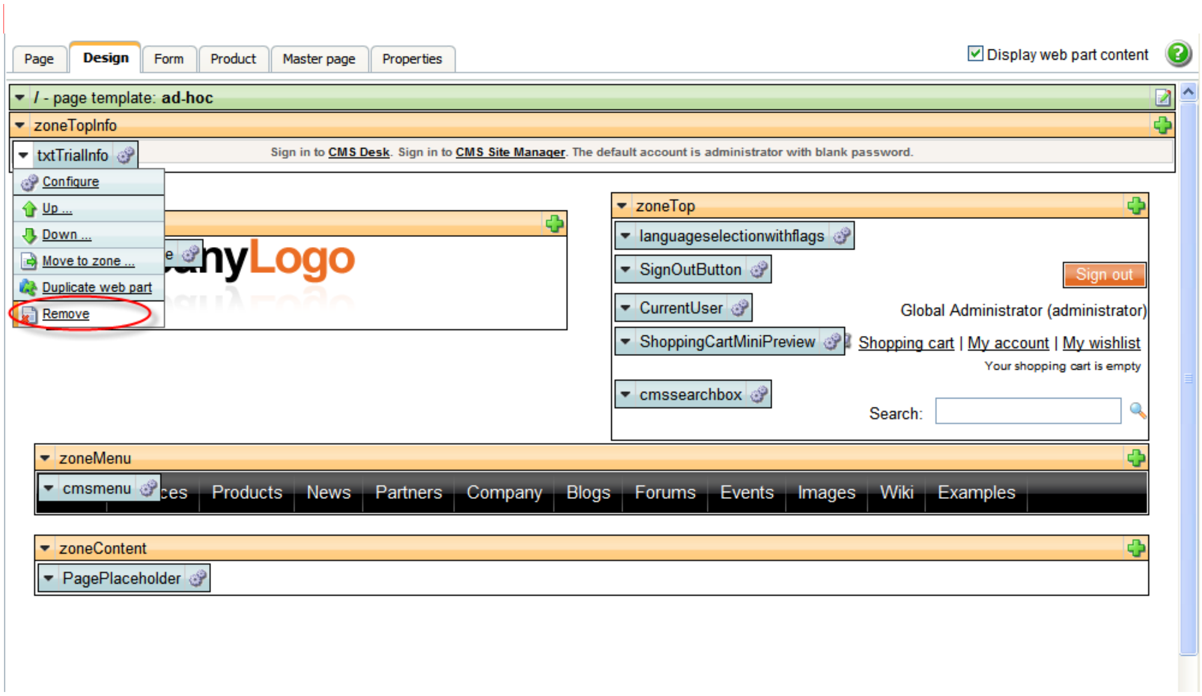
You've just publish a new logo on your web site.

## 67.1.2 Removing the log-on bar

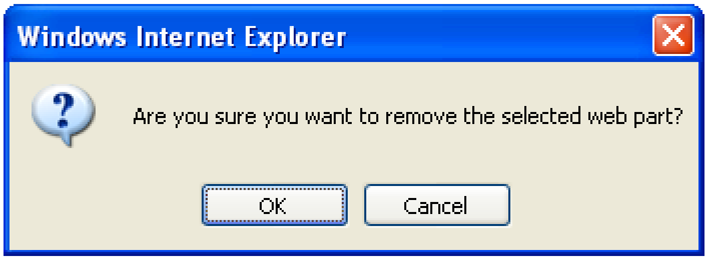
1. Go to **CMS Desk** -> **Content** -> **Corporate Site**.



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.



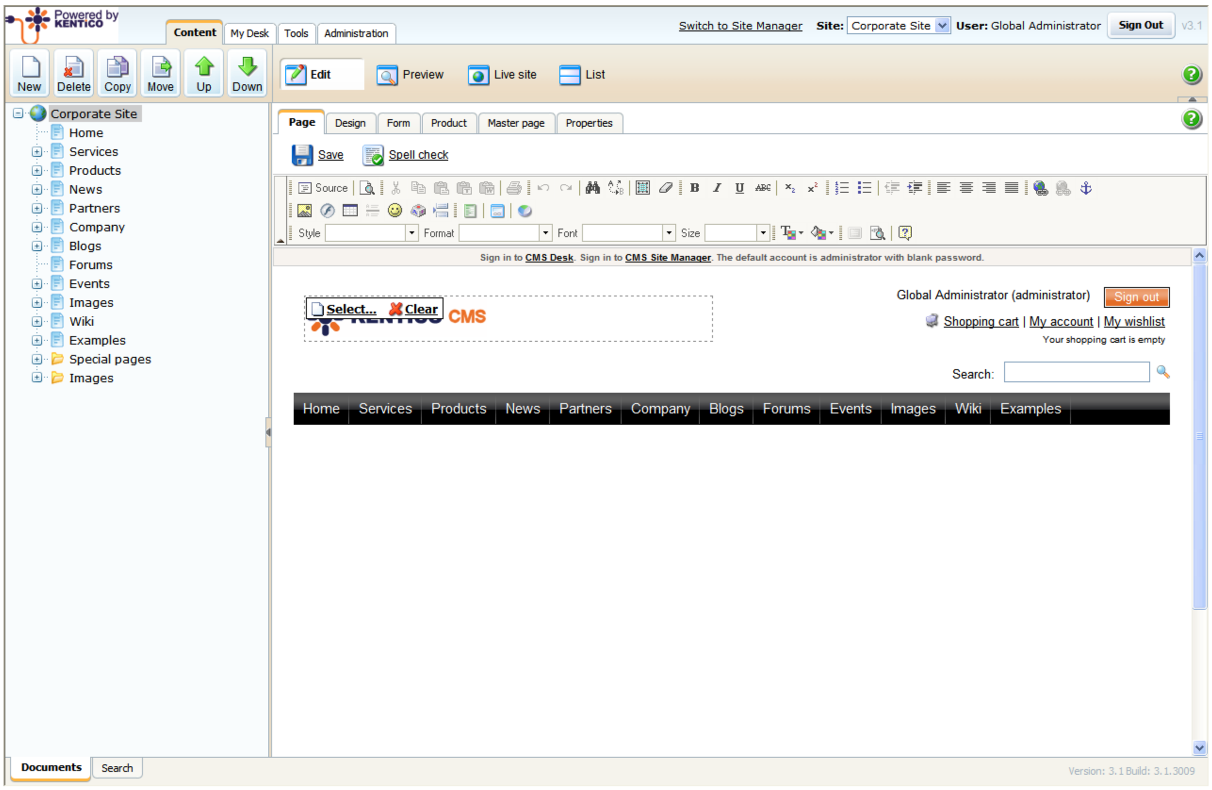

That's how you remove the signing-in bar from your web site.

## 67.1.3 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.

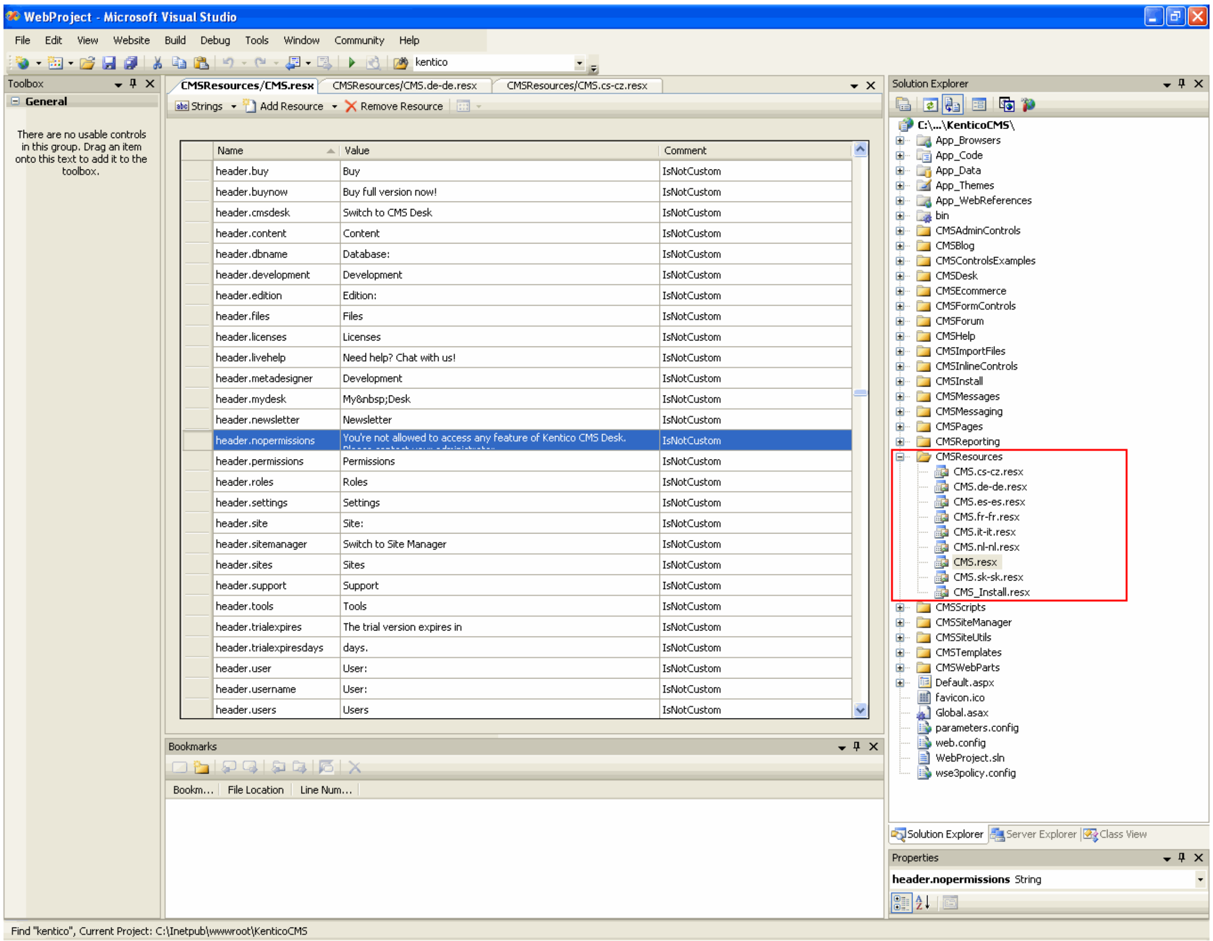
1. Go to **<your KenticoCMS directory>\App_Themes\Default\Images\CMSDesk** for changing the logo in CMS Desk or to **<your KenticoCMS directory>\App_Themes\Default\Images \CMSSiteManager** for changing the logo in CMS Site Manager.

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.



## 67.1.4 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.



# Part XX

# 68 Debugging & Optimalization

## 68.1 SQL queries debugging

SQL queries debugging can be turned on by adding the following keys to the *web.config* file of your web project:

This key will make the **SQL debug** tab visible in **Site Manager -> Administration -> System**. On this tab, you can see a log  of debugged pages SQL queries history.

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

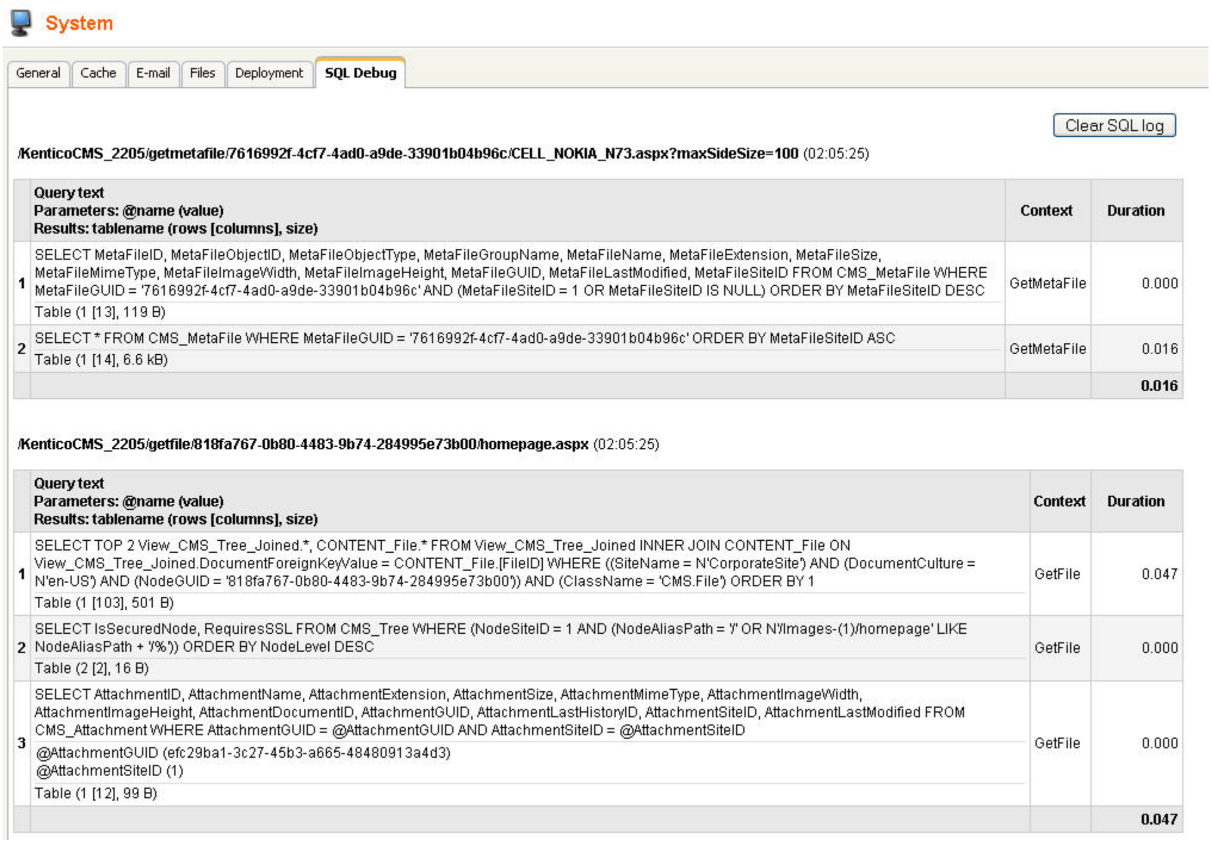
The value property of this key determines the number of last loaded pages displayed in the previously mentioned log.

```
<add key="CMSDebugSQLQueriesLogLength" value="10" />
```

You can see the appearance of the log in the screenshot below. 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.

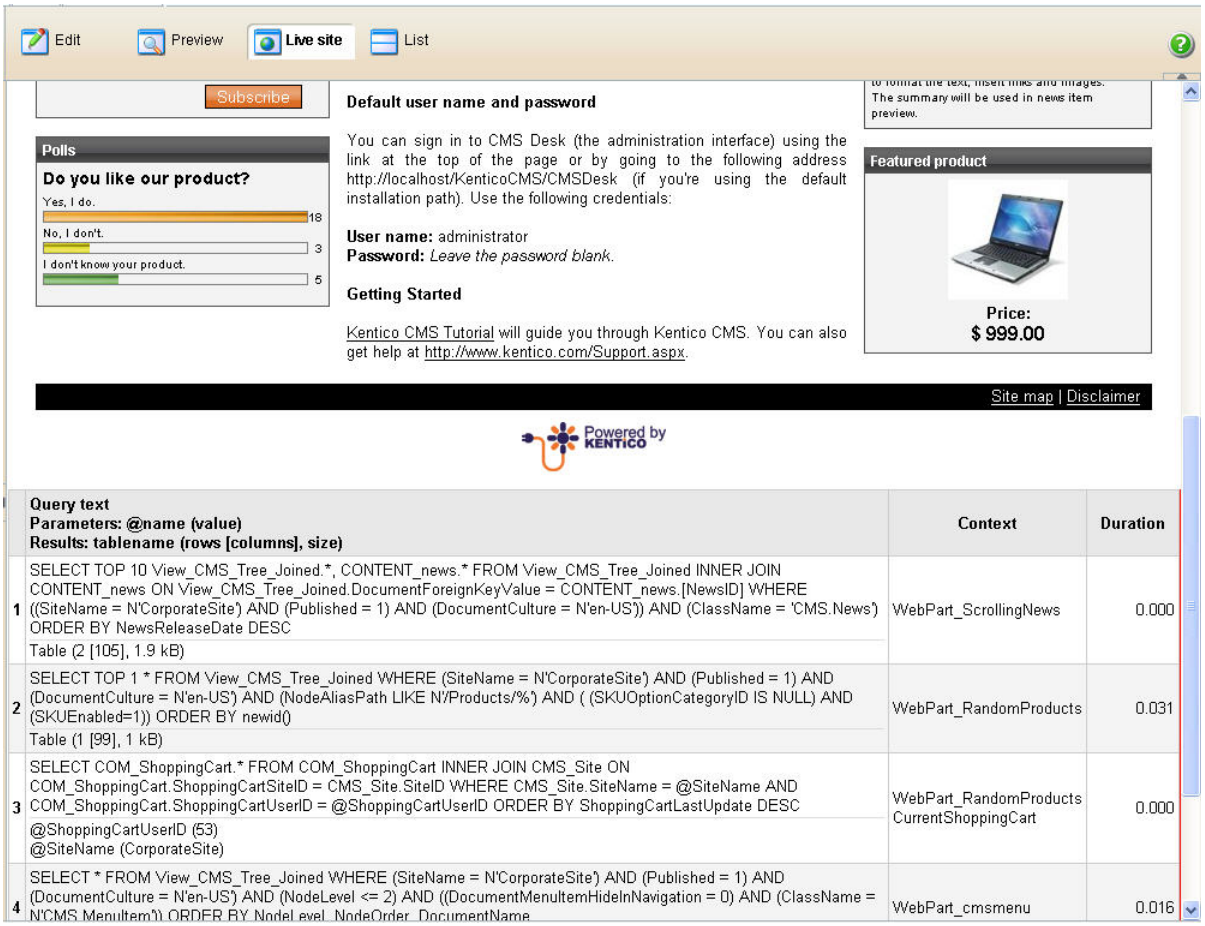


There are also different ways of displaying the query log:

The following key will turn on SQL debugging in **Live site** mode.

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

A log table of all SQL queries executed when loading a page will be displayed on the bottom of the page, as you can see in the following screenshot:



Adding this key will turn on logging SQL queries into a file. All executed SQL queries will be logged into *<web root>/AppData/logsql.log*:

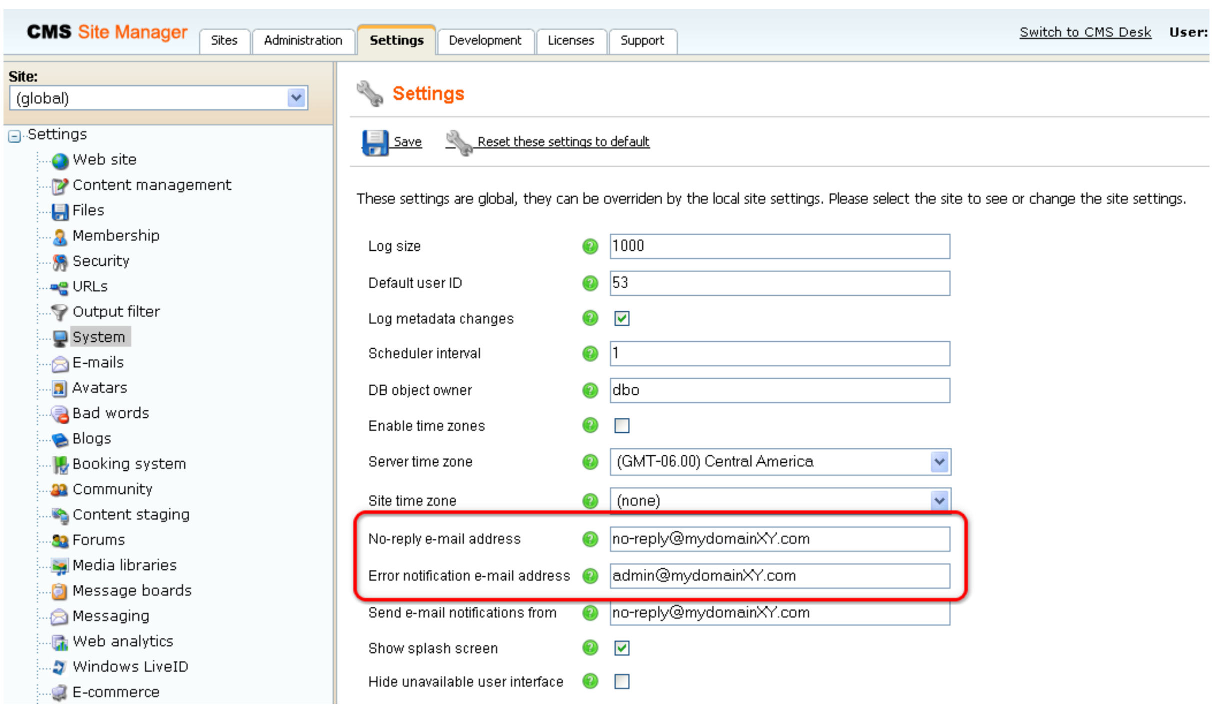
```
<add key="CMSLogSQLQueries" value="true" />
```

If you want to use SQL debugging on ASPX page templates, you need to manually add the following control to the page. It will display the same queries table as in the Live site mode debugging described above.

```
<cms:QueryLog ID="logQuery" runat="server" />
```

## 68.2 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.



# Part XX

# 69 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 often eliminates 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**><expression><parameters><**type**>}

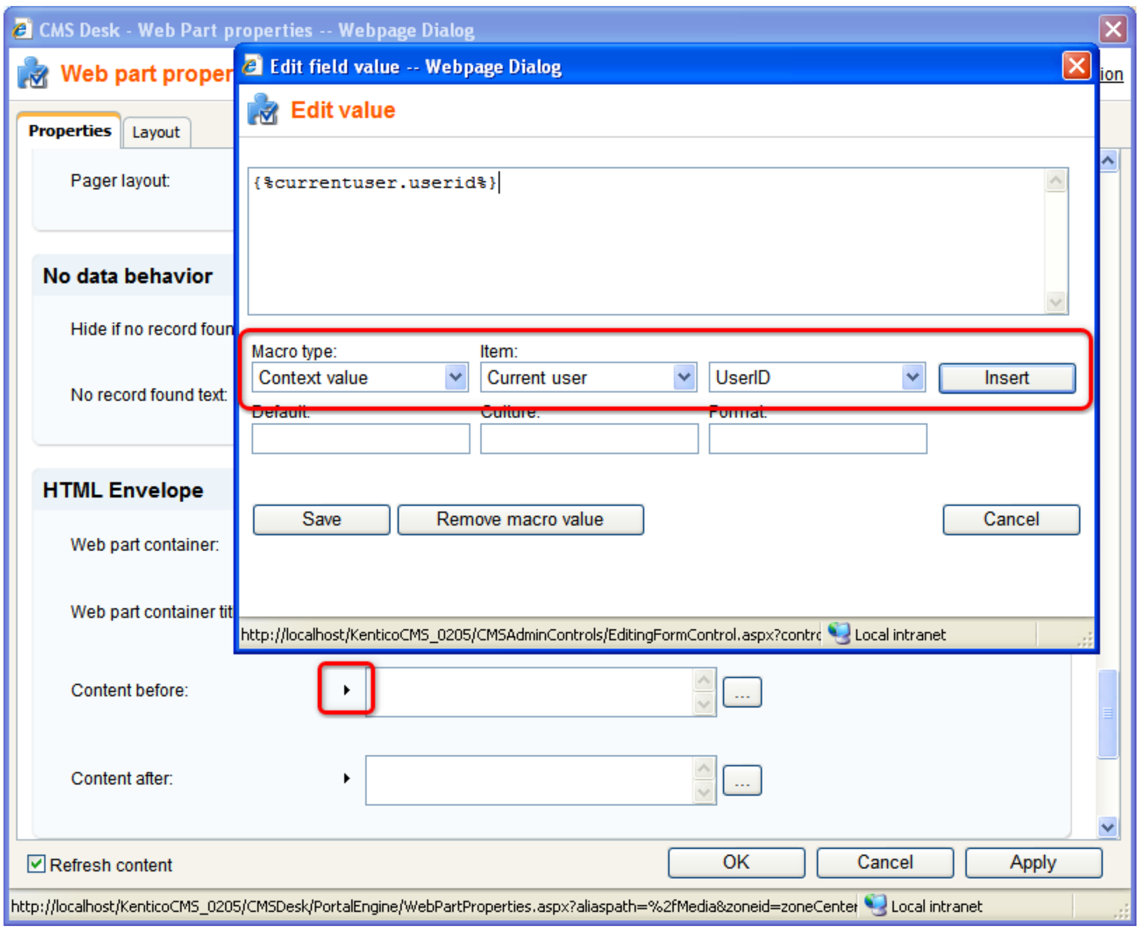
Where **type character** can be one of:

- **$** - Localization macro
- **%** - Context (data) macro
- **?** - QueryString macro
- @ - Cookie macro
- **#** - Custom macro

## Entering macros

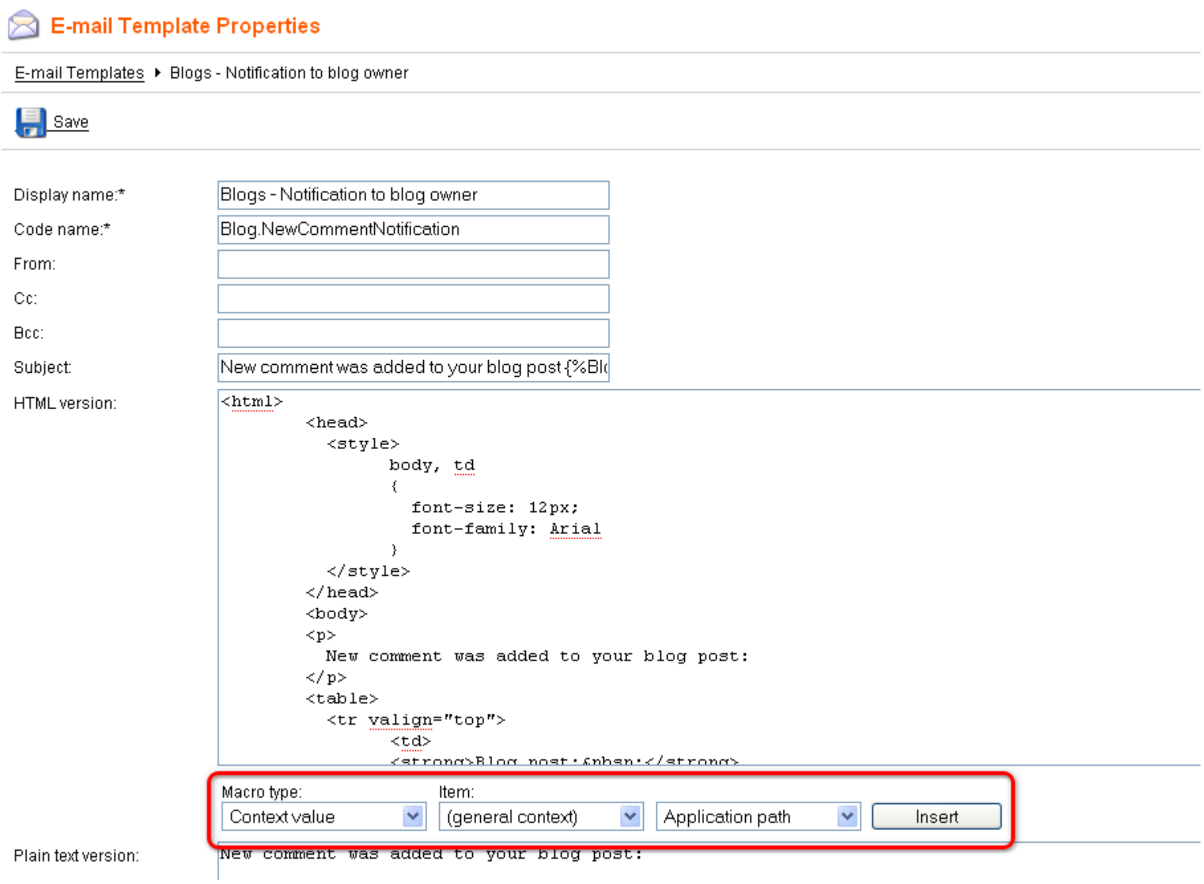
Macros can either be entered manually, or you can use the macro selector dialog, which facilitates entering of macro expressions and offers a complete range of macros to be entered.

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 three drop-down lists, inserted into the field above using the **Insert** button and finally added to the property using the **Save** button.



The same three drop-downs are also available when creating or editing:

- **E-mail templates**
- **Newsletter issues**
- **Newsletter templates**
- **E-commerce invoice templates**



## Localization macros

There are two types of locatization macros:

**Basic –** In format “{$string.key$}” uses the ResHelper.GetString(“string.key”) and replaces the macro with the resource string.

Example: “The weather is {$General.OK$}” will be resolved as “The weather is OK”

**In-place locatization –** String and its localized equivalents are stored within the macro specification. The macro is in format “{$=Default string|cs-cz=Czech string|de-de=German string$}”. It uses localized string in case it is available or default (the first string) if not.

Example:

“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
“The weather is OK” in any other culture

These macros are usually used on multilingual web sites to help localize the system strings.

## Context (data) macros

This type of macros evaluates the data of current context. The format of the macro is “{%ColumnName%}” and the value is replaced with the appropriate data from the context.

You can use all the column names from current site data, current user data and current document data.

Example: “Welcome {%FullName%}” will be resolved as “Welcome Global administrator” when administrator is logged in.

These macros can be used for example to parametrize the web parts parameters with current document or user values.

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. The macro is in format “{?querystringkey?}” and it is replaced by the query parameter value.

Example: “Current node ID: {?nodeid?}” will be resolved as “Current node ID: 10” with URL like “default.aspx?nodeid=10”

These macros can be used for example to dynamically parametrize the controls by the querystring parameters.

## Cookie macros

These macros evaluate the cookie values. The macro is in format "{@CookieName@}" and it is replaced by the cookie value.

Example: "Current style: {@StyleCookie@}" will be resolved as "Current style: Red" if StyleCookie value is set to "Red"

These macros can be used for example to parametrize the web parts with client-based persistent values like styles or user options.

## Custom macros

These macros can be used to define your own macro. The macro is in format "{#Expression#}" and when the macro is needed to be resolved, it will call the method **ResolveCustomMacro** located in the class ~/App_Code/Global/CMS/CMSCustom.cs

Example: "Current time: {#CurrentTime#}" will be resolved to "Current time: 1/1/2008 10:30" with following custom macro handler:

```
/// <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 the 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 can be used to resolve current document Alias path the same way like in the Path property of the controls. Macro is replaced by the resolved path.

Example: WhereCondition: “NodeAliasPath LIKE {&../%&}” The macro will be resolved as parent document path and will result in selecting all the siblings of 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.

## Macro parameters

From version 4.0 there is possibility to create macros with parameters to get better or specific functionality, especially for the 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}**%}

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>” – Processes 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 qurystring 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 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 comparisson

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 comparisson

e.g. **{%UserName|(equals)administrator|(truevalue)Yes|(falsevalue)No%}** writes No if the user is not administrator

## Coverage of macros

The following table shows where the macros can be used:

| | Locatization | Context | Query | Cookie | Custom | Path |
|---|---|---|---|---|---|---|
| **Web part properties** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **Web part containers** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **Page layouts** | Supported by API | | | | | |
| **Document metadata** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **Editable regions** | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| **Newsletters** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **Reporting** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **Forums** | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| **Polls** | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| **E-Commerce** | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ |
| **BizForms** | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| **Transformations** | Supported by API | | | | | |
| **.NET code** | Supported by API | | | | | |

## API methods for evaluating macros

There is an easy way to resolve the macros in the .NET code. To resolve all the macros (recommended), use static method:

```
string CMS.CMSHelper.CMSContext.CurrentResolver.ResolveMacros(string inputText)
```

To resolve just the localization macros use static method:

```
string CMS.GlobalHelper.ResHelper.LocalizeString(string inputText)
```

# Part XXX

# 70 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/webparts, such as CMSRepater/Document repeater or CMSDataGrid/Document datalist, 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