The system settings include *appSettings* keys and other settings, such as a connection string placed in appropriate sections of the project's **web.config** file. AppSettings keys are stored in the **/configuration/appSettings** section.

The following setting categories are available:

- General settings
- Active Directory settings
- Assembly settings
- Settings for forbidden characters in user and role names
- Settings for forbidden character replacement in URLs
- Content staging settings
- Continuous integration settings
- WYSIWYG editor settings
- Code editor settings
- E-commerce settings
- Email settings
- Event log settings
- Export/import settings
- Item listing settings
- URL settings for cultures
- Query string parameter name settings
- Scheduler settings
- Contact management settings
- Security settings
- Smart search settings
- Translation service settings
- User interface culture settings
- Web analytics settings
- Web farm synchronization settings
- Microsoft Azure settings
- Connection string

General settings

Key	Description	Default

CMSProgrammingLanguage	Indicates the programming language used in transformations and in custom code added to web parts.
	
	Note: Existing transformations are not automatically rewritten to the new language. We only recommend changing the value before starting the project's development phase.
	
	```<add key="CMSProgrammingLanguage" value="C#" />```
	
	or
	
	```<add key="CMSProgrammingLanguage" value="VB" />```
		C#

CMSTrialKey	Contains a temporary trial license key. You can remove this value after installation.	"" (empty string)

CMSHashStringSalt	Sets the salt value that the system uses in hash functions, for example when creating macro signatures. The installer automatically adds the key for new instances of Kentico, with a random GUID as the value. **Note**: Changing the hash salt on a website that already has defined content may prevent macros from resolving correctly, or break dialog links and images on your website. If you encounter such problems, you need to re-save the given content or globally re-sign macros (the system then creates the hashes using the new salt). `<add key="CMSHashStringSalt" value="eb2d6fac-8b9e-427c-b98b-3c562dffbb35" />`	Random GUID
CMSUseVirtualPathProvider	Indicates if the virtual path provider is enabled. The system uses the virtual path provider to load virtual objects, such as ASCX transformations and page layouts, from the database. `<add key="CMSUseVirtualPathProvider" value="false" />`	true
CMSRenderGeneratorName	Indicates if the 'generator' meta tag stating that the page was generated by Kentico is generated in the header of each page. `<add key="CMSRenderGeneratorName" value="true"/>`	false
CMSClearFieldEditor	Determines field editor behavior when creating new fields. If true, new fields will have empty values of attributes. If false, new fields will have predefined values, the same as the previously selected field. `<add key="CMSClearFieldEditor" value="true"/>`	true
CMSShowTemplateASPXTab	Indicates whether the **ASPX code** tab is visible when editing page templates. The tab allows you to export portal engine templates as ASPX page templates. `<add key="CMSShowTemplateASPXTab" value="true"/>`	false
CMSDatabaseCulture	Specifies the default culture of the system's database. `<add key="CMSDatabaseCulture" value="en-us"/>`	en-us
CMSDisposeConnectionAfterClose	If true, database connection is automatically disposed (allocated resources released) when a database connection is closed. `<add key="CMSDisposeConnectionAfterClose" value="true"/>`	false
CMSUseSessionCookies	Indicates whether session cookies are used. `<add key="CMSUseSessionCookies" value="true"/>`	true

CMSFileScriptTimeout	The maximum number of seconds a script can run before the server terminates it. ``` <add key="CMSFileScriptTimeout" value="7200"/> ```	7200
CMSUseExtensionOnPostback	When using extensionless URLs or custom extensions, certain legacy environments had problems processing postbacks. To prevent the postback problems, the system retains the ability to automatically attach the *.aspx* extension to the URL in the form tag. Set the key to *true* to attach the *.aspx* extension in postback URLs. ``` <add key="CMSUseExtensionOnPostback" value="false"/> ```	false
CMSUseTrailingSlashOnlyForExtensionLess	If the **Use URLs with trailing slash** setting is enabled in **Settings -> URLs and SEO**, this key determines whether the system adds a trailing slash only for URLs without an extension. For example, enabling the key ensures that URLs such as */googlesitemap.xml* do not have a trailing slash. ``` <add key="CMSUseTrailingSlashOnlyForExtensionLess" value="true"/> ```	false
CMSAllowCheckIOPermissions	If true, write permissions on the site folder are checked when necessary and produce an error message when they are insufficient. ``` <add key="CMSAllowCheckIOPermissions" value="true"/> ```	true
CMSWorkflowSendEmailToModerator	If true, workflow notification emails will be sent to the user who is performing the current workflow step along with other users involved in the workflow. ``` <add key="CMSWorkflowSendEmailToModerator" value="true"/> ```	false
CMSControlElement	If present in the web.config, the tag entered in the value will be used instead of the SPAN tag when generating pages. ``` <add key="CMSControlElement" value="div"/> ```	span
CMSUseParsedSelfClose	Indicates if parsed self-closing tags operations (faster) are used instead of standard self-close filter. ``` <add key="CMSUseParsedSelfClose" value="true"/> ```	true
CMSGetFileEndRequest	If true, *ApplicationInstance.CompleteRequest()* is used instead of *Response.End()* in the *CompleteRequest* method. ``` <add key="CMSGetFileEndRequest" value="true"/> ```	true

CMSDefault UserID	Specifies default user ID.	0
	```<add key="CMSDefaultUserID" value="53"/>```	
ImportFiles DiskPath	Specifies path to attachments that should be attached to pages imported via the SQL Import windows application.	"" (empty string)
	```<add key="ImportFilesDiskPath" value="C:\Temp"/>```	
CMSDelete Temporary Attachment sOlderThan	Specifies how old attachments need to be in order to be deleted by the **Delete old temporary attachments** scheduled task. The value is in hours. Attachments older than the entered value will be deleted when the scheduled task is executed.	24
	```<add key="CMSDeleteTemporaryAttachmentsOlderThan" value="12"/>```	
CMSDisable Administrat ionInterface	Allows you to disable the Kentico administration interface. The system displays an *Access denied* screen on each attempt to access the administration interface (~/Admin).	false
	```<add key="CMSDisableAdministrationInterface" value="true"/>```	

CMSMaxNo deAliasLen gth	**Warning**: We do NOT RECOMMEND using this web.config key. Scenarios with a non-default value are not tested or supported.
	
	Determines the maximum allowed length of page aliases. The default value is *50* characters and larger names are trimmed. You can increase the allowed length if you need to enter very long aliases (URLs).
	
	If you use the key, you also need to manually update your Kentico database:
	
	• Increase the size of the **NodeAlias** column in the *CMS_Tree* database table.
	• Change the size restriction in the *CMS.Tree* class definition in the **ClassXmlSchema** column of the *CMS_Class* table.
	
	```<xs:element name="NodeAlias" minOccurs="0"> ... <xs:restriction base="xs:string"> <xs:maxLength value="50" /> </xs:restriction> ... </xs:element>```
	
	• Update any dependent objects. (E.g., *View_CMS_Tree_Joined*.)
	
	Note: The maximum allowed length for the **Node alias path** is 450 characters, so the system may trim the alias path for pages deep in the content tree regardless of the allowed length.
	
	```<add key="CMSMaxNodeAliasLength" value="120" />```
		50

CMSMaxNodeNameLength	**Warning**: We do NOT RECOMMEND using this web.config key. Scenarios with a non-default value are not tested or supported.
	
	Sets the maximum possible length for the names of pages in the content tree. The default value is *100* characters and larger names are trimmed. This key can be used to increase the allowed length, which can be useful if you wish to enter very long page names.
	
	If you use the key, you also need to manually update your Kentico database:
	
	• Increase the size of the following database fields:
	   ○ **NodeName** column in the *CMS_Tree* table
	   ○ **DocumentName** column in the *CMS_Document* table
	   ○ **DocumentName** column in the *CMS_EventLog* table
	   ○ **PageTemplateDisplayName** column in the *CMS_PageTemplate* table
	   ○ **VersionDocumentName** column in the *CMS_VersionHistory* table
	• Change the size restriction of every column from the list above in its respective class (*CMS.Tree*, *CMS.Document*...) definition in the **ClassXmlSchema** column of the *CMS_Class* table.
	
	`<xs:element name="Name_of_the_field" minOccurs="0"> ... <xs:restriction base="xs:string"> <xs:maxLength value="100" /> </xs:restriction> ... </xs:element>`
	
	• Update any dependent objects (E.g., *View_CMS_Tree_Joined* and *IX_CMS_PageTemplate_PageTemplateCodeName_PageTemplateDisplayName*).
	
	`<add key="CMSMaxNodeNameLength" value="150" />`
		100

CMSWebAnalyticsSlidingIPExpiration	This key is used when the *Settings -> On-line marketing -> Web Analytics -> Remember visitors by IP (minutes)* key has a value higher than 0. If *enabled*, users who are active on the site but have disabled cookies are not logged as new visitors after the set time. If *disabled*, even an active user with disabled cookies is logged as a new site visitor after the set time. `<add key="CMSWebAnalyticsSlidingIPExpiration" value="false" />`	enabled
CMSAuthenticationType	This key overrides the values returned by the *IsFormsAuthentication* and *IsWindowsAuhentication* methods in *CMS.Helpers.RequestHelper*. You will typically use this if you are using a custom authentication provider whose authentication type is a non-standard one (e.g. Federated authentication) to make Kentico handle it as if it was windows or forms authentication. The following values are available: • **default**: standard behavior • **forms**: *IsFormsAuthentication* always returns *true*, *IsWindowsAuthentication* always returns *false* • **windows**: *IsFormsAuthentication* always returns *false*, *IsWindowsAuthentication* always returns *true* • **both**: *IsFormsAuthentication* always returns *true*, *IsWindowsAuthentication* always returns *true* `<add key="CMSAuthenticationType" value="windows" />`	default

CMSTemporaryFilesFolderPath	Overrides the default *~/App_Data/CMSTemp* location where various temporary files are stored by the system. As a value, you can use: • **physical disk path** - e.g. *c:\myfiles\mysite* • **virtual path** - e.g. *~/UploadedFiles* • **UNC path** - e.g. *\\server\folder* ``` <add key="CMSTemporaryFilesFolderPath" value=" \\server\MyCustomFolder" /> ```	
CMSDefaultTheme	Sets the name of the folder within *~\App_Themes* containing the theme that the system uses by default. ``` <add key="CMSDefaultTheme" value="MyTheme" /> ```	Default
CMSApplicationName	Used by Kentico Windows services to identify the Kentico instance. In case of IIS installations, the path to the instance in IIS is used as its value. In case of Visual Studio web server installation, the name of the target web project root folder is used. The value must be less than 60 characters long. It is also used to register Health monitoring performance counters and to identify respective counter categories when writing monitored values to them. ``` <add key="CMSApplicationName" value="Default Web Site/CMS" /> ```	
CMSApplicationGuid	Unique identifier of the Kentico instance. Used by Kentico Windows services to identify the Kentico instance. ``` <add key="CMSApplicationGuid" value="fe5493b6-4ea9-42e6-92fa- 0db1156b9163" /> ```	
CMSMediaFileMaxVersioningSize	Sets the maximal file size of versioned media files in kilobytes. Files in media libraries that are larger than the entered value will not have versions created on their update. ``` <add key="CMSMediaFileMaxVersioningSize" value="1024" /> ```	2147483647 (int. MaxValue)
CMSShowForgottenPassLink	Indicates if a link that allows users to recover a forgotten password should be displayed on the sign-in page to the administration interface. The default value is *true*. ``` <add key="CMSShowForgottenPassLink" value="false" /> ```	true

CMSSelectorMaxDisplayedItems	Sets the maximum amount of items displayed in dropdown list selectors in the administration interface when the number of selectable objects is higher than the value of the **CMSSelectorMaxDisplayedTotalItems** key. Users can select the remaining objects in a dialog that can be opened through the *(more items...)* option. The value of this key can be overridden for individual UniSelector controls. ```<add key="CMSSelectorMaxDisplayedItems" value="20" />```	25
CMSSelectorMaxDisplayedTotalItems	Limits the maximum number of items displayed in dropdown list selectors in the administration interface. If there are more selectable objects, the list is shortened (to the number specified by the **CMSSelectorMaxDisplayedItems** key) and the *(more items...)* option is added. The value of this key can be overridden for individual UniSelector controls. ```<add key="CMSSelectorMaxDisplayedTotalItems" value="30" />```	50
CMSWebAnalyticsShowFullData	Indicates whether the amount of data displayed in the analytics reports should be reduced before being rendered. The default value is *false* (for performance reasons). ```<add key="CMSWebAnalyticsShowFullData" value="true"/>```	false
CMSPhysicalFilesCacheMinutes	Sets the number of minutes for which clients consider physical files to be valid and up-to-date when using client caching (i.e. the length of the client cache expiration time for files). Specifically, the key determines how the server sets the **Expires** response header for file requests. The key only applies to files stored on the file system that are served using the *GetResource.ashx* handler. Other types of dynamic files use the expiration time set in **Settings -> System -> Performance -> Client cache (minutes)** within the administration interface. Physical files that are requested directly use the IIS client cache configuration. See also: Caching files and resources ```<add key="CMSPhysicalFilesCacheMinutes" value="60"/>```	10080
CMSHTMLEncodeEval	If true, the CMSAbstractTransformation class Eval methods used in ASCX transformations encode string values. ```<add key="CMSHTMLEncodeEval" value="true"/>```	false
CMSEmailValidationRegex	Allows you to customize the regular expression that the system uses to validate email addresses. ```<add key="CMSEmailValidationRegex" value="^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$"/>```	

CMSMacros CaseSensiti veComparis on	If true, string comparisons in macros are case sensitive (if not overriden by |*(casesensitive)* macro parameter). `<add key="CMSMacrosCaseSensitiveComparison" value="true"/>`	false
CategoryID Length	Category IDs in the CategoryIDPath column/field (the CMS_Category table) will be padded with leading zeros to the width specified in the key, e.g. 00001232. `<add key="CategoryIDLength" value="9"/>`	8
UIelementI DLength	UI element IDs in the ElementIDPath column/field (the CMS_UIElement table) will be padded with leading zeros to the width specified in the key, e.g. 00001232. The default value is *8*. `<add key="UIelementIDLength" value="9"/>`	8
SettingsCat egoryIDLen gth	Settings category IDs in the CategoryIDPath column/field (the CMS_SettingsCategory table) will be padded with leading zeros to the width specified in the key, e.g. 00001232. `<add key="SettingsCategoryIDLength" value="9"/>`	8
CMSDelete Temporary UploadFiles OlderThan	Specifies how old (in hours) unfinished upload files should be deleted by the **Delete old temporary upload files** scheduled task. `<add key="CMSDeleteTemporaryUploadFilesOlderThan" value="12"/>`	24
CMSImageE ditorMaxVe rsionsCount	Indicates how many versions of edited image should be saved in the image editor temporary folder. `<add key="CMSImageEditorMaxVersionsCount" value="500"/>`	999
CMSImageE xtensions	Specifies a list of file extensions which should be allowed as image files. `<add key="CMSImageExtensions" value=""/>`	""
CMSAudioE xtensions	Specifies a list of file extensions which should be allowed as audio files. `<add key="CMSAudioExtensions" value=""/>`	""
CMSVideoE xtensions	Specifies a list of file extensions which should be allowed as video files. `<add key="CMSVideoExtensions" value=""/>`	""

CMSUrlPort	Specifies the port number used in certain types of URLs (for example GetResource requests for images or modal dialog URLs). Standard navigation URLs of web pages are not affected. If not set, the URL port is automatically taken from the current request (80 in most cases). Add this key if you wish to generate URLs with a specific port number, regardless of the port supplied in requests. The CMSUrlPort key may also be useful in scenarios where requests are redirected to a nonstandard port number, but you wish to use the usual HTTP port in URLs (for example in a web farm environment, with a load balancer directing traffic to multiple servers identified by different port numbers). `<add key="CMSURLPort" value="80" />`	
CMSSSLUrl Port	Specifies the port number for certain types of URLs (for example GetResource requests for images or modal dialog URLs) for requests using the SSL (HTTPS) protocol. Overrides the **CMSUrlPort** key for SSL requests. Add this key if you wish to generate URLs with a specific port number, regardless of the port supplied in requests. `<add key="CMSSSLURLPort" value="443" />`	
CMSSocial MediaCooki eLevel	Sets the cookie level required to display the **YouTube video** web part and widget (due to the usage of third-party cookies). If your website uses a lower default cookie level than *All* (1000), YouTube videos will not be displayed for visitors unless they increase their allow cookie level. To enable YouTube videos for lower levels, set the corresponding cookie level value into this web.config key. `<add key="CMSSocialMediaCookieLevel" value="0" />`	1000
CMSDefault PageClassN ame	Use the key to change the default value of the page menu item classname. `<add key="CMSDefaultPageClassName" value="CMS.MenuItem" />`	CMS. MenuItem

CMSXUACompatibleValue	Defines a value of X-UA-Compatible header meta tag. This meta tag allows you to choose the version of Internet Explorer, in which the page should be rendered. Available options are: • IE=5 • IE=EmulateIE7 • IE=7 • IE=EmulateIE8 • IE=8 • IE=EmulateIE9 • IE=9 • IE=edge - uses the highest mode available for the particular version of Internet Explorer. The meta tag is added if IE browser is used. `<add key="CMSXUACompatibleValue" value="IE=9" />`	IE=edge
CMSUseXUACompatible	Defines if X-UA-Compatible header meta tag can be added. `<add key="CMSUseXUACompatible" value="false" />`	true
CMSDisplayApplicationErrorMessages	Defines the level of information which is displayed to the user if the application fails to initialize (e.g., the application fails to connect to the SQL server): • false - only general error message is displayed. • true - detailed information about the origin of the error is displayed. `<add key="CMSDisplayApplicationErrorMessages" value="true" />`	false
CMSUpdateLastLogonForExternalUsers	Indicates whether the system updates the last sign-in time when authenticating external users. For example, you can set the key to *false* to avoid user updates on each request when using Windows authentication. `<add key="CMSUpdateLastLogonForExternalUsers" value="false" />`	true
CMSTransactionIsolationLevel	Sets the Isolation level used for queries that are executed as transactions. Use only in special cases if you encounter problems with deadlocks. For more information about the available values, see: IsolationLevel Enumeration `<add key="CMSTransactionIsolationLevel" value="ReadUncommitted" />`	ReadCommitted

CMSUseAbsoluteCSSClientURLs	If set to *true*, the system converts URLs inside CSS stylesheets from relative to absolute (including the protocol and site domain) when serving the actual CSS code to browsers. For example, *background:url(/media/images/pic.png)* becomes *background:url(http://www.domain.com/media/images/pic.png)*. Only enable the key if you need to ensure backward compatibility for stylesheets from previous Kentico versions that rely on the absolute URL conversion. `<add key="CMSUseAbsoluteCSSClientURLs" value="true" />`	false
CMSForceAttachmentHistoryInDatabase	Allows you to overwrite the *Settings -> System -> Files -> Store files in* setting for page attachment history and store the data in the database instead of the file system. `<add key="CMSForceAttachmentHistoryInDatabase" value="true" />`	false
CMSPortalTemplatePage	Indicates which page is used as a base template for portal engine pages. Use this key with caution as improper use may lead to non-functional website. Creating a modified template page and using this key is the preferred way of customizing the template. Directly modifying the *~/CMSPages/PortalTemplate.aspx* file is strongly not recommended. `<add key="CMSPortalTemplatePage" value="~/CMSPages/CustomPortalTemplate.aspx">`	~/CMSPages/PortalTemplate.aspx

> Back to list of web.config key categories

Active Directory settings

You can use the following keys to configure the import of users and roles when using Windows Active Directory authentication:

Key	Description	Default
CMSImportWindowsRoles	When Windows authentication is used and this key set to true, the domain groups of users in Active Directory are imported into the system as roles. `<add key="CMSImportWindowsRoles" value="false"/>`	true
CMSRoleImportFormat	When the mixed mode authentication is used, this key determines the format of role names which are imported from Active Directory: • **SAM** - the format of imported roles is *rolename*. • **DomainSAM** - the format of imported roles is *domain-rolename*. `<add key="CMSRoleImportFormat" value="DomainSAM"/>`	DomainSAM

| CMSADDefaultMapUserName | When the mixed mode authentication is used, this key determines the format of user names which are used to communicate with the Active Directory provider:

• **sAMAccountName** - uses the Active Directory user name (without the domain).
• **userPrincipalName** - uses the Active Directory user principal name (e.g. *MyName@office.example.com*).

<pre><add key="CMSADDefaultMapUserName" value="userPrincipalName" /></pre> | sAMAccountName |

> Back to list of web.config key categories

Assembly settings

You can use the following keys to assign custom assemblies to be used instead of certain default assemblies:

Key	Description	Default

CMSStorageProviderAssembly	Specifies the assembly that implements a custom <u>file system provider</u> for the local Windows file system.
	
	<pre><add key="CMSStorageProviderAssembly" value="Company.
	FileSystemStorage"/></pre>
		-

CMSSearchProviderAssembly	Specifies the assembly that implements a custom version of the full-text search based on the SQL engine.
	
	<pre><add key="CMSSearchProviderAssembly" value="Company.
	SearchProviderSQL" /></pre>
		-

CMSDataProviderAssembly	Specifies the assembly that implements a custom data provider assembly used as the database connector.
	
	See <u>Custom Data provider example</u>.
	
	<pre><add key="CMSDataProviderAssembly" value="Company.
	DataProvider"/></pre>
		-

CMSVirtualPathProviderAssembly	Specifies the assembly that implements a custom <u>Virtual path provider</u>.
	
	The system uses the virtual path provider to load virtual objects, such as ASCX <u>transformations</u> and <u>page layouts</u>, from the database.
	
	<pre><add key="CMSVirtualPathProviderAssembly" value="Company.
	VirtualPathProvider"/></pre>
		-

> Back to list of web.config key categories

Settings for forbidden characters in user and role names

You can use the following keys to configure forbidden characters in user and role names:

Key	Description	Default
CMSEnsureSafeUserNames	Indicates if the system replaces forbidden characters in user names imported from Active Directory. If the key is *false,* the system also uses less restrictive validation of users names for standard users (applies when new users register, or when creating or editing users in the administration interface). See the *CMSUserValidationRegEx* key for details. `<add key="CMSEnsureSafeUserNames" value="false" />`	true
CMSEnsureSafeRoleNames	Indicates if the system replaces forbidden characters in the code names of roles imported from Active Directory. If the key is *false,* you may experience problems with roles names containing certain forbidden characters, for example with leading underscores characters (_). Therefore, it is NOT RECOMMENDED to set the key to false. `<add key="CMSEnsureSafeRoleNames" value="false" />`	true
CMSForbiddenUserNameCharactersReplacement	Sets the character that replaces forbidden characters in user names imported from Active Directory. If not set, the value from *Settings -> URLs and SEO -> Forbidden characters replacement* is used. `<add key="CMSForbiddenUserNameCharactersReplacement" value="-" />`	
CMSForbiddenRoleNameCharactersReplacement	Sets the character that replaces forbidden characters in role names imported from Active Directory. If not set, the value from *Settings -> URLs and SEO -> Forbidden characters replacement* is used. `<add key="CMSForbiddenRoleNameCharactersReplacement" value="-" />`	

CMSUserValidatio nRegEx	Sets a custom regular expression that determines which user names are valid. Applies when new users register on the website, are created in the administration interface or through the API, and when the name of an existing user is modified.	14
	The default value is: ^[a-zA-Z0-9_\-\.@]+$	
	If the *CMSEnsureSafeUserNames* key is set to *false*, a less restrictive regular expression is used by default: ^[a-zA-Z0-9_\-\.\\@!#$%&'*+/=?^_`{\|}~]+$	
	Note: The key does not apply when processing the names of external users, and does not affect replacement of invalid characters in the names of users imported from Active Directory.	
	```<add key="CMSUserValidationRegEx" value="([A-Za-z0-9-]+)" />```	

> Back to list of web.config key categories

Settings for forbidden character replacement in URLs

You can use the following keys to configure how forbidden URL characters should be replaced:

Key	Description	Default
CMSForbid denURLVal ues	The characters entered as the value of this key are forbidden in URLs (page aliases and URL paths) and will be replaced automatically by the character specified in the **Forbidden characters replacement** setting in **Settings -> URLs and SEO**. By default, the following characters are forbidden: \/:*?"<>|&%.'#[]+=„" and the space character. If you add this key to the web.config, its value will override the default forbidden character set. This way, you can either allow some of the forbidden characters or add new ones. Please note that it is recommended to keep the default characters forbidden, since they may prevent certain types of URLs from working correctly if entered into URL paths. Also, keep in mind that some characters need to be escaped in the web.config according to standard XML rules, e.g. enter *<* instead of the < character. ```<add key="CMSForbiddenURLValues" value="\/:$?"<>|&%.'#[] =" />```	
CMSLimitUr lReplaceme nts	While enabled, consecutive forbidden characters in URLs will be replaced by only a single replacement character and forbidden URL characters located at the beginning or end of the path will be removed completely instead of being replaced. If set to *false*, each forbidden character is replaced individually. ```<add key="CMSLimitUrlReplacements" value="false" />```	true

CMSUseLimitReplacementsForUrlPath	Indicates if the functionality enabled by the **CMSLimitUrlReplacements** key should be applied to the **Page URL Path** property. `<add key="CMSUseLimitReplacementsForUrlPath" value="false" />`	true

> [Back to list of web.config key categories](#)

Content staging settings

You can use the following keys to configure the behavior of [Content staging](#):

Key	Description	Default
CMSStaging AcceptAllCe rtificates	If set to *true,* the system accepts all certificates when performing content staging tasks over a secured protocol (SSL). Includes self-signed certificates. If *false,* the system only accepts certificates generated by a trusted certification authority. `<add key="CMSStagingAcceptAllCertificates" value="true" />`	false
CMSMediaF ileMaxStagi ngSize	Sets the maximum size of [media library files](#) that are synchronized via staging (in kiloBytes). Add the key to the web.config file of projects that work as **source servers** in your staging environment. The system does not synchronize binary data for media library files above the limit. Synchronization tasks for such files only transfer the database definitions of the media objects, without the actual files. `<add key="CMSMediaFileMaxStagingSize" value="1024" />`	2147483 647 (int. MaxValu e)
CMSStaging TreatServer NamesAsIns tances	On Kentico instances that contain multiple sites and use staging, changes made to global objects produce staging tasks for all staging servers across all sites. If you synchronize such tasks at different times on different sites, you risk overwriting newer data with the data of old synchronization tasks (we recommend synchronizing such changes at the same time on all servers and sites). If you have full control over all sites on your instance, you can define the same set of target servers for each site (each server must have an identical *Code name* across all sites), and set the *CMSStagingTreatS erverNamesAsInstances* key to *true*. When a global staging task is synchronized, the system then automatically deletes the task from servers with matching code names on the other sites. The default value is *false* because code name collisions may occur for staging servers on unrelated sites. `<add key="CMSStagingTreatServerNamesAsInstances" value="true" />`	false
CMSStaging ServerName	Name of the staging server used for advanced bi-directional content staging. The value needs to be used as the code name when defining this server as a target server of other servers. See [Bi-directional content staging](#) for more information. `<add key="CMSStagingServerName" value="server2" />`	"" (empty string)

CMSStaging ServiceTim eout	Sets timeout interval for the service used to synchronize staging tasks. The key must be configured within the web.config file on the source instance from which the tasks are being synchronized. Enter the value in seconds. `<add key="CMSStagingServiceTimeout" value="240" />`	180
CMSSynchr onizeShare dTemplates WithDocum ents	Indicates if changes made to page templates used by multiple pages should be synchronized together with page update synchronization tasks of all pages using the template. If disabled, page update tasks will not include page template synchronization tasks. `<add key="CMSSynchronizeSharedTemplatesWithDocuments" value="false" />`	true
CMSStaging UseTreeCus tomHandlers	If true, events from the _DocumentEvents_ class are raised when processing page staging tasks. `<add key="CMSStagingUseTreeCustomHandlers" value="false" />`	true
CMSStaging LogGlobalO bjectsOnlyF orAssigned Sites	If true, tasks for global objects are logged only for the sites to which the respective objects are assigned. `<add key="CMSStagingLogGlobalObjectsOnlyForAssignedSites" value="true" />`	false
CMSStaging UseAutoma ticOrdering	Determines whether the system applies automatic ordering to pages created via staging. If _true_ (by default), the system overrides the ordering of new pages according to the **Content -> Content management -> New page ordering** setting. Configure this web.config key on **_target_** servers that process incoming staging tasks. `<add key="CMSStagingUseAutomaticOrdering" value="false" />`	true

> Back to list of web.config key categories

Continuous integration settings

You can use the following keys to configure the continuous integration solution:

Key	Description	Default

CMSCIEncoding	Sets the character encoding used when the continuous integration solution generates non-binary files in the project's **CIRepository** folder. The key supports values matching the encoding names listed in the Encoding Class article.	utf-8
	Changing the value of the *CMSCIEncoding* key does not update the encoding type of existing files in the *CIRepository* folder. To fully update the encoding of the repository content, you need to:	
	• Run complete serialization for all objects. • Manually update the encoding type of your *repository.config* file.	
	```<add key="CMSCIEncoding" value="utf-16" />```	

> Back to list of web.config key categories

WYSIWYG editor settings

You can use the following keys to configure the WYSIWYG editor:

Key	Description	Default
CKEditor:DefaultToolbarSet	Allows you to set the default toolbar when you use the 'default' option for toolbar sets on a page. You can specify one of the following values: • Full • Standard • Basic You can make custom toolbar sets 'default' the same way. Note that the values are case sensitive. ```<add key="CKEditor:DefaultToolbarSet" value="Full" />```	Standard
CMSWYSIWYGFixXHTML	Indicates if the WYSIWYG editor should automatically try to fix XHTML incompatibilities in the code it generates. Supported values are *true* and *false*. ```<add key="CMSWYSIWYGFixXHTML" value="true" />```	true
CKEditor:BasePath	Specifies the location of the WYSIWYG editor (CKEditor). By default, it is located in *~/CMSAdminControls/CKeditor*. ```<add key="CKEditor:BasePath" value="/CKeditor/" />```	

| CKEditor: PersonalizeToolbarOnLiveSite | Indicates if the CK toolbar can be personalized on the live site.

Supported values are *true* and *false*.

```
<add key="CKEditor:PersonalizeToolbarOnLiveSite" value="true" />
``` | false |

> Back to list of web.config key categories

Code editor settings

You can use the following keys to configure the Code editor, which is used in the interface to ensure syntax highlighting and work with code fields:

Key	Description	Default
CMSEnableSyntaxHighlighting	Globally enables or disables the advanced editor and syntax highlighting support for all code fields. This can be used to turn off the editor if it is causing performance issues or other problems. ``` <add key="CMSEnableSyntaxHighlighting" value="false" /> ```	true
CMSEnableSyntaxHighlighting. <Language>	Can be used to disable the advanced editor and syntax highlighting support for fields that display code in a specific language. Replace the *<Language>* string with the name of the language that you wish to disable. The following language options are available: **Text**, **HTML**, **CSS**, **JavaScript**, **XML**, **CSharp**, **SQL**, **HTMLMixed**, **ASPNET**, **CMSSharp**. All languages are enabled by default. ``` <add key="CMSEnableSyntaxHighlighting.CSS" value="false" /> ```	
CMSShowLineNumbers	Determines whether the code editor in the administration interface displays the panel containing line numbers by default. **Note**: The editor fields used in the Kentico administration interface always have line numbers disabled by default. The key only affects custom fields that have displaying of line numbers enabled (fields using the *Macro editor* form control with the *Show line numbers* parameter enabled). ``` <add key="CMSShowLineNumbers" value="false" /> ```	true

> Back to list of web.config key categories

E-commerce settings

You can use the following keys to configure the Kentico E-commerce Solution:

Key	Description	Default

CMSUse MetaFil eForPro ductIm age	Indicates if meta files should be used for product images. If set to **true**, the system asks the <u>user</u>, while creating or editing a <u>product</u> (or a <u>product option</u>), to select a product image from the file system on their computer. The system then uploads the product image to a server and saves it as a product metafile (a product image page of the **cms.file** type is not created). The path to this metafile is saved as the product image path (*SKUImagePath*). If set to **false**, the system asks the user to select a page of the **cms.file** type as a product image. The path to the selected page is saved as the product image path (*SKUImagePath*). ```<add key="CMSUseMetaFileForProductImage" value="true"/>```	true

> Back to list of web.config key categories

Email settings

You can use the following keys to configure the sending of emails:

Key	Description	Default
CMSEmail TransferE ncoding	Specifies the type of transfer encoding used for emails sent from Kentico. Supported values are: *base64, quotedprintable, sevenbit* ```<add key="CMSEmailTransferEncoding" value="QuotedPrintable" />```	base64
CMSEmail SendLimit	Sets the maximum number of messages sent to individual SMTP servers in a single connection. When the limit is reached, the connection is automatically closed and a new one opens. The process continues until all messages are sent. You can lower the value if you encounter SMTP errors with too many messages set per connection. ```<add key="CMSEmailSendLimit" value="20" />```	50
CMSLogE mails	Enables logging of all sent emails to the **~/App_Data/logemails.log** file. The log contains each email's timestamp, recipients, sender, subject, and custom headers and their encoding (UTF-8 by default). ```<add key="CMSLogEmails" value="true"/>```	Depends on debugging settings

CMSDebu gEmails	Disables sending emails to the actual recipients. The system only logs emails into the event log. Helpful if you need to test the functionality, but do not want the emails to actually reach the recipients. • To view the event log, open the **Event log** application. • The emails are logged as **Information** type events. • The **Event code** column contains the recipient's address. • The system randomly generates *Sending failed for <recipient's email address>* errors to simulate sending errors. ```<add key="CMSDebugEmails" value="true"/>```	false

> Back to list of web.config key categories

Event log settings

You can use the following keys when you need to achieve specific behavior of the Event log:

Key	Description	Default
CMSLog Events	Indicates if logging of events in the Event log is enabled. ```<add key="CMSLogEvents" value="true"/>```	true
CMSLog KeepPer cent	Coefficient for Event log deletion. Keeps the specified percentage of extra log items in the log with regards to the **Settings -> System -> Event log size** setting. The specified percentage of the oldest events is deleted by batch when the percentage is exceeded. If 0, the exact number of records is kept in the log. ```<add key="CMSLogKeepPercent" value="10"/>```	10
CMSLog EventsTo File	If true, events are also logged into the ~*App_Data\logevents.log* file. ```<add key="CMSLogEventsToFile" value="true"/>```	false
CMSLogF ieldChan ges	If true and the **Settings -> System -> Log metadata changes** option is enabled, details about particular object changes are included in the respective log records. ```<add key="CMSLogFieldChanges" value="true"/>```	false
CMSLog Docume ntFieldC hanges	If true and the **Settings -> System -> Log metadata changes** option is enabled, details about changes of values in page fields are included in the respective log records. ```<add key="CMSLogDocumentFieldChanges" value="true"/>```	false

CMSLog Newsletter IssueE vents	Indicates if the system logs marketing email changes into the event log (including logging of all sent issues). You can disable logging of these events if you encounter problems with performance or a cluttered event log when mailing issues to a very large number of recipients – subscribed contacts. `<add key="CMSLogNewsletterIssueEvents" value="false" />`	true
CMSLog MATransi tions	Indicates if the system logs all step transitions in Marketing automation processes into the event log. You can disable logging of these events if you encounter performance issues or a cluttered event log when running automation processes for a very large number of contacts. `<add key="CMSLogMATransitions" value="false" />`	true
CMSData ExportTe mplateF older	Indicates the starting path for template lookup. `<add key="CMSDataExportTemplateFolder" value="~/App_Data/CMSModules /DataExport"/>`	~ /App_Data /CMSModu les /DataExpo rt

> Back to list of web.config key categories

Export/import settings

You can use the following keys when you want to configure file exporting:

Key	Description
CMSExportExcluded Folders	Specifies which folders are filtered from the Files folder of the export package. **.svn** folders are excluded by default, even without this key added. `<add key="CMSExportExcludedFolders" value="test*;cms*" />`
CMSExportExcluded Files	Specifies which files are filtered from the Files folder of the export package. **.scc** files are excluded by default, even without this key added. `<add key="CMSExportExcludedFiles" value="test*;cms*" />`

CMSSiteUtilsFolderP ath	Overrides the default *~/CMSSiteUtils/* location where the system stores export and import packages by default. As a value, you can use: • **physical disk path** - e.g. *c:\myfiles\mysite* • **virtual path** - e.g. *~/UploadedFiles* • **UNC path** - e.g. *\\server\folder* `<add key="CMSSiteUtilsFolderPath" value="\\server\MyCustomFolder" />`

> Back to list of web.config key categories

Item listing settings

You can use the following keys to configure item listing in the Kentico administration interface:

> ⚠ **Note**: The web.config keys listed below only define default global settings, and individual listing components (UniGrid controls) can override the values. As a result, the values that you configure may not affect all parts of the administration interface.

Key	Description	Default
CMSDefaul tListingFilt erLimit	Determines the minimum number of items that must be included in a listing in order for a filter to be shown. If the number of listed items is lower than this value, the filter is not displayed. If it is larger, the filter is displayed. This applies to all listings (UniGrid controls) across the entire UI. `<add key="CMSDefaultListingFilterLimit" value="40" />`	25
CMSDefaul tPageSizeO ptions	Sets the options available in the page size selector for listings in the Kentico interface (the *Items per page* setting). Values must be separated by commas. You can use the *##ALL##* value to add an option that lists all available rows on a single page (not recommended if you have objects with a very large number of records). `<add key="CMSDefaultPageSizeOptions" value="10,50,200" />`	5,10,25, 50,100
CMSDefaul tListingPag eSize	Sets the initial page size of listings in the Kentico interface (the *Items per page* setting). If the value is not included among the options defined by the **CMSDefaultPageSizeOptions** key, the system automatically inserts the value as an additional option. `<add key="CMSDefaultListingPageSize" value="50" />`	25
CMSListing ShowFirstL astButtons	If true, the first and last page link buttons are included in the pagers of listings in the administration interface (with a large enough number of items). If false, the buttons are always hidden. `<add key="CMSListingShowFirstLastButtons" value="true" />`	false

CMSListing ShowDirect PageContr ol	If true, a text box that allows the current page to be changed by directly entering a number is included in pagers of listings in the administration interface (with a large enough number of items). If false, the control is always hidden. `<add key="CMSListingShowDirectPageControl" value="false" />`	true

> Back to list of web.config key categories

URL settings for cultures

You can use the following keys to set up URL behavior for the culture (language) versions of pages:

Key	Description	Default
CMSUseCultureFor BestPageInfoResult	Indicates whether the currently selected culture should have the highest priority when deciding which language version of a page should be displayed. The default value is *false*, which means that accessing a page through the custom URL set for one of its specific culture versions will override and change the preferred culture accordingly. If set to *true*, the currently selected culture will be reflected even when a culturespecific page URL path is used. `<add key="CMSUseCultureForBestPageInfoResult" value="true" />`	false

> Back to list of web.config key categories

Query string parameter name settings

You can use the following keys to change certain query string parameter names:

Key	Description	Default
CMSLanguageP arameterName	Changes the name of the query string parameter used to set the culture. For example, this allows you to get *Home?sprache=de-de* instead of the default *Home?lang=de-de*. `<add key="CMSLanguageParameterName" value="sprache" />`	lang
CMSAliasPathP arameterName	Changes the name of the *aliaspath* query string parameter so that you get *products.aspx?ap= /products/myproduct* instead of the default *products.aspx?aliaspath=/products/myproduct*. `<add key="CMSAliasPathParameterName" value="ap" />`	aliaspath

> Back to list of web.config key categories

Scheduler settings

By adding the following keys to your web.config, you can configure the scheduler:

Key	Description	Default

CMSUseAutomaticScheduler	Indicates if the system uses automatic scheduling. When enabled, the scheduler periodically requests the **~/CMSPages/Scheduler.ashx** route, which ensures that scheduled tasks are processed regularly even if there is no website activity. If disabled (*false* - by default), tasks are processed at the end of standard page requests. ```<add key="CMSUseAutomaticScheduler" value="true" />```	false
CMSRunSchedulerWithinRequest	If *true* (the default value), the scheduler is executed within the standard **EndRequest** event of a page. If *false*, the scheduler is executed via the *~/CMSPages/Scheduler.ashx* route. ```<add key="CMSRunSchedulerWithinRequest" value="false" />```	true
CMSSchedulerURL	Sets the URL used by the automatic scheduler. Must target the *~/CMSPages/Scheduler.ashx* route or the URL of a fully custom scheduler. ```<add key="CMSSchedulerURL" value="https://domain/cmspages/scheduler.ashx" />```	~/CMSPages/Scheduler.ashx
CMSSchedulerAcceptAllCertificates	If true, all security certificates (including not valid ones) will be accepted when accessing the *~/CMSPages/Scheduler.ashx* route via a secured protocol. ```<add key="CMSSchedulerAcceptAllCertificates" value="true" />```	false
CMSSchedulerUserName	Sets the user name under which the *~/CMSPages/Scheduler.ashx* route is accessed (e.g. when using windows authentication). ```<add key="CMSSchedulerUserName" value="office\myname" />```	"" (blank username)
CMSSchedulerPassword	Sets the password for the user name under which the *~/CMSPages/Scheduler.ashx* route is accessed. ```<add key="CMSSchedulerPassword" value="mypassword123" />```	"" (blank password)

> Back to list of web.config key categories

Contact management settings

By adding the following keys to your web.config, you can configure the behavior of contact management.

Key	Description	Default
CMSProcessContactActionsInterval	Sets the interval (in seconds) in which contact activities are batch processed by the system. Consider raising the interval if your system processes a large number of activities. ```<add key="CMSProcessContactActionsInterval" value="30" />```	10

CMSCreateContact ActionsLogWorker	If enabled, all web farm servers recalculate contact scores, contact groups, personas, and marketing automation trigger s. If you wish to improve performance, you can set this key to *false* for one or more of your web farm servers, and the recalculation will be disabled for them. **Note**: If you set the key value to *false* for all of your web farm servers, the recalculation will be disabled completely. `<add key="CMSCreateContactActionsLogWorker" value="false"/>`	true

> Back to list of web.config key categories

Security settings

By adding the following keys to your web.config, you can configure security options:

Key	Description	Default
CMSAcce ptAllCerti ficates	Indicates whether application requests should accept all certificates (including invalid certificates). This key ensures the same as if the following keys were enabled at the same time: • *CMSSearchCrawlerAcceptAllCertificates* • *CMSSchedulerAcceptAllCertificates* • *CMSStagingAcceptAllCertificates* `<add key="CMSAcceptAllCertificates" value="true"/>`	false
CMSAdmi nEmerge ncyReset	Allows you to recover access to the administration interface if you forget the password of your administrator account. The value must consist of the following parts, separated by semicolons: • Username of the account • New password for the account • Optional true/false value which specifies whether you want to create a new user with the Global administrator privilege level. The system automatically deletes the key after you gain access to the administration interface. See also: Forgotten password `<add key="CMSAdminEmergencyReset" value="admin;password;true" />`	
CMSAllo wOnlySi mpleMac ros	If true, only simple macros (i.e. those which do not need a security check) are allowed. All others will not be resolved. If true, CMSTextBox control does not add security signatures to macros. `<add key="CMSAllowOnlySimpleMacros" value="true"/>`	false

CMSEnableCsrfProtection	Indicates whether the system uses security tokens to protect pages against [Cross site request forgery](#) attacks. Only set this key to *false* if you have your own security mechanism against CSRF. ```<add key="CMSEnableCsrfProtection" value="false" />```	true
CMSEnableWIF	Overwrites the setting of the **Enable WIF authentication** option in **Settings -> Security & Membership -> Authentication -> Claims-based authentication**. If included in the web.config file and set to false, this key can **override enabled WIF authentication** in case you have not established an administrator account (as described in [Claims-based authentication](#)) and you do not have access to the administration interface. ```<add key="CMSEnableWIF" value="false"/>```	
CMSPasswordSalt	The password salt is a string that is appended to user passwords before they are hashed (to improve security). By default, the salt contains the randomly generated GUID of the given user. The content of this key is added after the base salt value to further increase the length of the salt. Only applies if the application stores passwords using the *SHA2 with salt* format, which you can configure in **Settings -> Membership & Security -> Passwords -> Password format**. See also: [Setting the user password format](#) ```<add key="CMSPasswordSalt" value="SaltText" />```	
CMSRenewSessionAuthChange	When set to *true,* the system enforces a change of the session ID when a user signs in or out. Provides protection against session fixation attacks. If you enable this key, users cannot preserve their session data after signing in or out. See also: [Session protection](#) ```<add key="CMSRenewSessionAuthChange" value="true" />```	false
CMSXFrameOptionsExcluded	Excludes pages under the specified paths from clickjacking protection. For pages under the excluded path, the system no longer adds the ***X-frame-options: SAMEORIGIN*** header to the HTTP response (this header ensures that pages displayed in frames must originate on the same server as the parent page). To exclude multiple paths, enter values separated by semicolons (;). See also: [Clickjacking protection](#) ```<add key="CMSXFrameOptionsExcluded" value="/Services" />```	
CMSUserBasedFloodProtection	Indicates whether the flood protection mechanisms are based on user accounts. ```<add key="CMSUserBasedFloodProtection" value="true"/>```	true

[> Back to list of web.config key categories](#)

Smart search settings

You can use the following keys to configure the Smart search:

Key	Description	Default
CMSSearchIndexPath	Sets the path of a custom directory where the system stores smart search index files (only applies to local search indexes). `<add key="CMSSearchIndexPath" value="App_Data\MyCustomIndexes" />`	
CMSCrawlerAllowSiteAliasRedirect	By default, Page crawler search indexes read pages that use redirection from the site's main domain name to a domain alias. Setting this key to *false* only allows indexing for pages that use the website's main domain. `<add key="CMSCrawlerAllowSiteAliasRedirect" value="false" />`	true
CMSCreateTemplateSearchTasks	If enabled, any changes made to page templates automatically trigger an update of all pages that are based on the given template in the appropriate smart search indexes. The default value is *true*. You can set this key to **false** if you wish to improve the performance or save resources in scenarios where you have a very large number of pages on your website that share the same page template. This way, the system no longer performs bulk updates of pages in the search index whenever their template is modified. Disabling this key means that your page index will not reflect changes to the page template (e.g. if you add static text to the template through a web part) until the given pages are updated for another reason, or the entire index is rebuilt. The key only affects changes to page templates. Editing the content of editable regions on a specific page always causes an update of the search index content. `<add key="CMSCreateTemplateSearchTasks" value="false" />`	true
CMSProcessSearchTasksByScheduler	By default, the smart search creates and runs local indexing tasks immediately whenever content covered by a search index is created or modified. If you set this key to *true*, the system does NOT run indexing tasks upon creation — they need to be processed periodically, typically using the **Execute local search tasks** scheduled task. **Note**: The key does not affect processing of tasks for Azure search indexes. Azure search tasks are always processed by the **Execute Azure search tasks** scheduled task. `<add key="CMSProcessSearchTasksByScheduler" value="true" />`	false

CMSSearch CrawlerAcc eptAllCertif icates	By default, the smart search crawler does not index pages on sites using a secured protocol (HTTPS) without a certificate from a trusted authority. For example, you may encounter the problem in development environments that use a self-signed certificate. If you set this key to *true*, the system accepts all certificates. **Note**: Setting the *CMSSearchCrawlerAcceptAllCertificates* key to *true* poses a security vulnerability and should not be used in production environments. `<add key="CMSSearchCrawlerAcceptAllCertificates" value="true" />`	false
CMSSearch MaxFieldLe ngth	Sets the maximum number of terms that the smart search allows in a single index field. `<add key="CMSSearchMaxFieldLength" value="10000" />`	Unlimited
CMSSearch MaxResults Number	Sets the maximum number of results that the smart search can return for a single search request (when using local search indexes). If a search produces more matching results, the system ignores the items that do not fit into the limit. The purpose of the limit is to avoid processing of large numbers of search results with low relevance. If you use very large search indexes that regularly produce more than *1000* results (the default limit), you can either increase the value or set up search filters to help narrow down the search. `<add key="CMSSearchMaxResultsNumber" value="3000" />`	1000
CMSSearch Synonyms Weight	Sets the relevance (result score) that the search assigns to items found through the synonym search (i.e. when using **Any word or synonyms** search mode). The key's value must be a decimal number ranging from 0 to 1. A larger number assigns higher relevance to synonyms. If you set 1, the score of synonyms is equal to words in the original unexpanded search expression. Only applies when using local search indexes. `<add key="CMSSearchSynonymsWeight" value="0.8" />`	0.9
CMSSmartS earchLockP ollInterval	Specifies how often (time in milliseconds) the smart search indexer attempts to acquire a lock on index files (for local indexes). The default value of this key is 500 milliseconds. Set a lower value if the following exception is being logged in your Event log. *Lock obtain timed out: CMS.SiteProvider.SearchLock* `<add key="CMSSmartSearchLockPollInterval" value="300" />`	500 ms

CMSSmartSearchLockTimeout	Defines the timeout period (in milliseconds) during which the smart search indexer attempts to acquire a lock on index files (for local indexes). The default value of this key is 1000 milliseconds. ```<add key="CMSSmartSearchLockTimeout" value="1500" />```	1000 ms
CMSSubsetAnalyzerWordRegex	Sets a [regular expression](#) that determines which characters are allowed in the "words" processed by [local search indexes](#) with **Subset** and **Starts with** analyzers. The analyzers use the following steps to create search tokens: 1. Divide text into words 2. Tokenize the substrings inside the words (according to the analyzer type) The words created in the first step may contain only characters that match the regular expression in the key's value. Any other characters split the text into separate words. By default, the following characters are allowed: • word characters (upper and lower case letters, numbers, underscores) • at symbols (@) • periods (.) **Note** : After changing the value of the key, you need to **Rebuild** your search indexes that use *Subset* or *Starts with* analyzers. ```<add key="CMSSubsetAnalyzerWordRegex" value="(\w|@|\.|\$)+" />```	(\w|@|\.)+
CMSSearchStoreContentField	Determines whether the smart search stores the original untokenized Content field within index files. **Warning**: Setting the key to *true* reduces indexing performance and increases the size of index files (the difference can be GBs of data for very large indexes). Only enable the key if you need to inspect or debug the content of your index files on development sites. Never enable the key on production sites. ```<add key="CMSSearchStoreContentField" value="true" />```	false

The following keys adjust how the smart search indexes page attachments (see [Searching attachment files](#)):

Key	Description	Default

CMSSearchDetectTextEncoding	Indicates whether the attachment search uses automatic detection of character encoding for text files (**txt** and **csv**). By default, the search can read text files that use the following encoding: • UTF-8 • The default Windows encoding (the operating system's current ANSI code page) If you enable automatic detection, the system attempts to detect the encoding type for each file, and use the correct option when reading the content. **Note**: Correct encoding detection is not guaranteed for all files. Automatic detection also slightly increases the time required to index text files. `<add key="CMSSearchDetectTextEncoding" value="true" />`	false
CMSSearchIndexXmlAttributes	Determines whether the attachment search includes the values of tag attributes when indexing XML files. `<add key="CMSSearchIndexXmlAttributes" value="true" />`	false
CMSSearchIndexXmlComments	Determines whether the attachment search includes comments when indexing XML files. `<add key="CMSSearchIndexXmlComments" value="true" />`	false
CMSSearchMaxAttachmentSize	Sets the maximum allowed file size in kB for the attachment search. The search ignores page attachments whose size exceeds the value. Indexing very large files can be resource intensive and have a negative impact on your website's performance. `<add key="CMSSearchMaxAttachmentSize" value="10000" />`	0 (unlimited)

> Back to list of web.config key categories

Translation service settings

You can use the following key to configure Translation services:

Key	Description	Default
CMSTranslationServicesUseCDATAForTransUnit	Use if you want to export XLIFF files without the CDATA notation. `<add key="CMSTranslationServicesUseCDATAForTransUnit" value="false" />`	true

CMSTranslations UseAbsoluteLinks	Determines whether the URLs in content sent for translation are converted to absolute format. If true, the links in content sent for translation are absolute and not converted back to relative format when the translated content is imported back to the system. Note that using absolute links in content sent for translation can cause issues when, for example, synchronizing the imported translation to a different domain by staging. `<add key="CMSTranslationsUseAbsoluteLinks" value="true" />`	false

> Back to list of web.config key categories

User interface culture settings

By adding the following keys to your web.config, you can configure the cultures used in the administration interface:

Key	Description	Default
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" />`	en-us
CMSShowLogonCultureSelector	Indicates if the user interface sign-in page should display a drop-down list with available user interface languages. `<add key="CMSShowLogonCultureSelector" value="false" />`	true
CMSDefaultUICulture	Specifies the default UI culture. If you use this key, you also need to: 1. Add a corresponding resource string file (localization pack) into the ~*CMSResources* folder. 2. Rename the default CMS.resx file to *CMS.en-us.resx*. 3. Rename the new culture's resource string file to *CMS.resx*. This is needed because the *CMS.resx* file is used when the *(default)* option is selected as a user's *Preferred user interface culture*. `<add key="CMSDefaultUICulture" value="en-nz"/>`	en-us
CMSUseSQLResourceManagerAsPrimary	Changes the priority of used localization resource strings to: 1. custom.resx 2. cms.resx 3. database (Kentico administration -> Localization application) `<add key="CMSUseSQLResourceManagerAsPrimary" value="false" />`	true

> Back to list of web.config key categories

Web analytics settings

By adding the following key to your web.config, you can configure Web analytics.

Key	Description	Default
CMSWebAnalyticsRemoveReferralQuery	Indicates whether URL query strings are removed when logging *Referrals* and *Referring sites* in web analytics. When false, the system then no longer aggregates *Referrals* and *Referring sites* from the same source with different query string parameters into a single URL. `<add key="CMSWebAnalyticsRemoveReferralQuery" value="false"/>`	true

> Back to list of web.config key categories

Web farm synchronization settings

By adding the following keys to your web.config, you can enable or disable web farm synchronization of certain kind of files stored in the file system:

Key	Description	Default
CMSWebFarmMode	Indicates if web farms are: • **disabled** • **automatic** - no further configuration is needed. • **manual** - requires manual configuration of web farm servers in the system. `<add key="CMSWebFarmMode" value="automatic"/>`	disabled
CMSWebFarmServerName	Code name of the web farm server. This value is used for manual web farm synchronization support. This setting overrides the name assigned by the system in the Automatic web farm mode. `<add key="CMSWebFarmServerName" value="server1" />`	"" (empty string)
CMSWebFarmMaxFileSize	If the **CMSWebFarmSynchronizeFiles** key is enabled, you can limit the maximum size of synchronized files using this key. The value is entered in kiloBytes and files larger than this value will not be synchronized. `<add key="CMSWebFarmMaxFileSize" value="1024" />`	2147483647 (int. MaxValue)
CMSWebFarmApplicationPhysicalPath	Path to the application on the disk used for synchronizing physical files. `<add key="CMSWebFarmApplicationPhysicalPath" value="C:\inetpub\wwwroot\Kentico" />`	

CMSWebFarmSync Interval	Average time before a created web farm synchronization task is processed by web farm servers (the maximum possible time before a task is processed is double of the specified interval). `<add key="CMSWebFarmSyncInterval" value="100" />`	2000 ms
CMSWebFarmSync hronizeFiles	Determines if web farms synchronize files in general. This key enables synchronization of: • Attachments • Meta files • Media files • Form files • Avatars • Forum attachments **Microsoft Azure notes**: The key is *false* by default for projects running in Microsoft Azure Cloud Services. For projects hosted in Microsoft Azure Web Apps, we strongly recommend that you manually set the key to *false*. Otherwise, you may encounter file system collisions when scaling to multiple instances. `<add key="CMSWebFarmSynchronizeFiles" value="false" />`	true
CMSWebFarmSync hronizeAttachmen ts	Enables/disables synchronization of attachments. `<add key="CMSWebFarmSynchronizeAttachments" value="true" />`	true
CMSWebFarmSync hronizeMetaFiles	Enables/disables synchronization of meta files. `<add key="CMSWebFarmSynchronizeMetaFiles" value="true" />`	true
CMSWebFarmSync hronizeMediaFiles	Enables/disables synchronization of media files. `<add key="CMSWebFarmSynchronizeMediaFiles" value="true" />`	true
CMSWebFarmSync hronizeBizFormFil es	Enables/disables synchronization of form files. `<add key="CMSWebFarmSynchronizeBizFormFiles" value="true" />`	true
CMSWebFarmSync hronizeAvatars	Enables/disables synchronization of Avatars. `<add key="CMSWebFarmSynchronizeAvatars" value="true" />`	true

CMSWebFarmSync hronizeForumAtta chments	Enables/disables synchronization of forum attachments. ``` <add key="CMSWebFarmSynchronizeForumAttachments" value="true" /> ```	true
CMSWebFarmSync hronizeDeleteFiles	Enables/disables synchronization of deleted files. ``` <add key="CMSWebFarmSynchronizeDeleteFiles" value="true" /> ```	true

> Back to list of web.config key categories

Microsoft Azure settings

By adding the following keys to your **web.config** file, you can set up the deployment of your website to Microsoft Azure and configure its behavior.

Note that there is an easier way of editing the definition end configuration files of your Azure project in Visual Studio by right-clicking the appropriate role in **CMSAzure -> Roles** and selecting **Properties** . For more information, refer to Adding application settings in an Azure project .

Key	Description
CMSAzureProject	Must be set to *true* if you wish to run the application in Microsoft Azure Cloud Services. *True* by default if the application is installed as a Microsoft Azure project, *false* in standard installations. ``` <add key="CMSAzureProject" value="true"/> ```
CMSAzureAccoun tName	Specifies the name of the Azure Storage account that the application will use for its file system. If you wish to run the application on the local emulator, enter *devstoreaccount1* as the value. ``` <add key="CMSAzureAccountName" value="devstoreaccount1"/> ```
CMSAzureShared Key	Contains the primary access key for the storage account specified in the CMSAzureAccountName setting. You can find the appropriate value for your storage account on the Microsoft Azure Management Portal. ``` <add key="CMSAzureSharedKey" value=" Eby8vdM02xNOcqFlqUwJPLlmEtlCDXJ1OUzFT50uSRZ6IFsuFq2UVErCz4I6tq/K1SZFPTOtr /KBHBeksoGMGw=="/> ```
CMSAzureBlobEn dPoint	Sets the endpoint used for the connection to the blob service of the specified storage account. If you wish to use the default endpoint, clear the value of this setting. ``` <add key="CMSAzureBlobEndPoint" value="http://127.0.0.1:10000 /devstoreaccount1"/> ```

CMSAzureQueueEndPoint	Sets the endpoint used for the connection to the queue service of the specified storage account. If you wish to use the default endpoint, clear the value of this setting. ```<add key="CMSAzureQueueEndPoint" value="http://127.0.0.1:10001/devstoreaccount1"/>```
CMSAzureTableEndPoint	Sets the endpoint used for the connection to the table service of the specified storage account. If you wish to use the default endpoint, clear the value of this setting. ```<add key="CMSAzureTableEndPoint" value="http://127.0.0.1:10002/devstoreaccount1"/>```
CMSAzureRootContainer	Specifies the name of the blob container that will serve as the root of the application's file system on the Azure storage account. This can be useful in scenarios where multiple applications use the same storage account. The default value is *cmsstorage*. ```<add key="CMSAzureRootContainer" value="CustomRoot"/>```
CMSAzurePublicContainer	Indicates if the blob container used to store the applications file system should be public. If set to *true*, it will be possible to access files directly through the URL of the appropriate blob service, for example: *<StorageAccountName>.blob.core.windows.net/cmsroot/corporatesite/media/imagelibrary/logo.png* ```<add key="CMSAzurePublicContainer" value="true"/>```
CMSAzureStorageCacheCleanerThreshold	Specifies a threshold in gigabytes that is used by the **Clean Azure storage cache** scheduled task. When the threshold is exceeded and the scheduled task is enabled, the system starts to delete files in the **Azure Cache** and **Azure Temp** folders from the oldest to the newest. The default value is *45GB*. ```<add key="CMSAzureStorageCacheCleanerThreshold" value="45"/>```
CMSAzureStorageCacheCleanerKeepLimit	Specifies an amount in gigabytes that is used by the **Clean Azure storage cache** scheduled task . The amount is the target size of the **Azure Cache** and **Azure Temp** folders when the scheduled task deletes the files. The size of the excluded paths (set by the **CMSAzureStorageCacheCleanerExcludedPaths** web.config key) is included in the amount. The default value is *10GB*. ```<add key="CMSAzureStorageCacheCleanerKeepLimit" value="10"/>```

CMSAzureStorageCacheCleanerExcludedPaths	Specifies relative paths separated by semicolons (;) that are excluded when using the **Clean Azure storage cache** scheduled task . There is not any predefined excluded path by default. ```<add key="CMSAzureStorageCacheCleanerExcludedPaths" value="folder_name1; folder_name2"/>```
CMSAzureCDNCacheMinutes	Allows you to set caching interval for Azure CDN. Using this key, you can set different intervals for expiration of files in the server memory (which can be set in the Settings application -> System -> Performance -> Cache files (minutes) option) and for expiration of files in Azure CDN. If this key is not used, the **Cache files (minutes)** setting is used for CDN expiration interval instead. See also Settings - Performance. ```<add key="CMSAzureCDNCacheMinutes" value="30"/>```

If you wish to host your website onpremise, but use a file system based on the **Blob Storage** service of an Azure storage account, you can specify the following settings:

Key	Description
CMSExternalStorageName	Configures the application to use an external storage provider for its *entire* file system. Do NOT set this key if you only wish to map specific folders to a different storage provider. See Configuring file system providers to learn how to implement such scenarios. The following key values are available for the external storage providers supported by default: • Azure Blob storage - **azure** • Amazon S3 - **amazon** ```<add key="CMSExternalStorageName" value="azure" />```
CMSAzureTempPath	The folder specified by this key will be used to store temporary files on a local disk, e.g. when transferring large files to or from the storage account. Do not use this key if the entire application is deployed as a Microsoft Azure hosted service. ```<add key="CMSAzureTempPath" value="C:\AzureTemp" />```
CMSAzureCachePath	Specifies a folder on a local disk where files requested from the storage account will be cached. This helps minimize the amount of blob storage operations, which saves time and resources. Do not use this key if the entire application is deployed as a Microsoft Azure hosted service. ```<add key="CMSAzureCachePath" value="C:\AzureCache" />```

CMSDownl oadBlobTi meout	Specifies the timeout interval in minutes for importing files from Azure Blob storage into Kentico. The default value is *1.5* minutes. Increase the interval if you encounter problems when importing large (about 2GB) files. ``` <add key="CMSDownloadBlobTimeout" value="50" /> ```

> Back to list of web.config key categories

Connection string

The database used by Kentico is specified by the connection string **CMSConnectionString** in the **/configuration/connectionStrings** section.

```
<add name="CMSConnectionString" connectionString="Persist Security Info=False;
database=CMS;server=myserver;user id=sa;password=mypassword123;Current Language=English;
Connection Timeout=120;" />
```