



General

Enable output compression	<p>Enables compression for the HTML output code of all pages rendered by Kentico.</p> <p>Output compression requires compliance from client browsers. Browsers that do not support compression always download uncompressed page output.</p> <p>Applies globally for all sites in the system.</p>
---------------------------	---

Server content caching

Cache page info (minutes)	<p>Sets the number of minutes for which the system caches page information (basic page properties and metadata). Kentico retrieves page information many times during the processing of a single page, so always set this value to at least 10 minutes.</p> <p>When a page is modified, the system automatically clears the corresponding part of the page info cache. This type of caching cannot cause the website to display outdated information.</p>
Cache content (minutes)	<p>Sets the number of minutes for which web parts/controls with data sources cache their content (typically retrieved from the Kentico database).</p> <p>You can override this value for specific web part instances by setting their Cache minutes property. Using 0 as the value disables content caching for the given web part instance.</p> <p>It is recommended to cache all possible content. You can use cache dependencies to clear the cache on content changes. For most non-custom data sources, the default dependencies automatically ensure that web parts reload the cached content whenever the data is modified.</p> <p>See also: Caching the data of page components</p>
Use progressive caching	<p>If checked, the system optimizes access to uncached data so that concurrent threads only use a single data access operation and share the results. This leads to better performance if the website is under a heavy load, without the drawback of not having the latest data available.</p>

Server file caching

Cache files (minutes)	<p>Sets the number of minutes for which the system caches images and other files on the server. Includes images, and web resources such as CSS stylesheets and JavaScript files.</p> <p>The system automatically removes files from the cache if they are modified, so file caching cannot cause the website to display outdated content.</p> <p>This setting also specifies the caching interval for Azure CDN, unless overwritten by the <i>CMSAzureCDNCacheMinutes</i> web.config key (see Reference - Web.config application keys for details).</p> <p>See also: Caching files and resources</p>
Maximum file size to cache	<p>Limits caching of files whose size is greater than the specified number of kilobytes. Files larger than the specified value are only cached for 1 minute, regardless of the <i>Cache files (minutes)</i> setting.</p>



Redirect files to disk	<p>If enabled, the system redirects file requests for page attachments and content tree files to the corresponding physical file (if the requested file is stored in the file system).</p> <p>Warning: The system stores files on the file system with GUID identifiers instead of names (to avoid conflicts). If you enable the <i>Redirect files to disk</i> setting, files downloaded by users have names in the GUID identifier format instead of the original file name.</p>
Client caching	
Client cache (minutes)	<p>Sets the number of minutes for which clients consider cached files to be valid and up-to-date (i.e. the length of the client cache expiration time). Specifically, the setting determines how the server sets the Expires response header for file requests. Client file caching includes images and web resources such as CSS stylesheets.</p> <p>The following values are possible:</p> <ul style="list-style-type: none"> • 0 - cached files are always flagged as expired, so clients revalidate files on each request. Does NOT disable client caching for files. • More than 0 - until a cached file expires, clients load the file's data from the cache without sending any requests to the server. Provides the best file performance, but may cause browsers to display outdated content (if cached files are modified before the expiration interval ends). <p>This setting only applies to dynamic files that are managed by the Kentico application, such as stylesheets in the <i>CSS stylesheets</i> application. Files stored on the file system may use a different client cache expiration time. For example, physical files served by the Kentico application through the <i>GetResource.ashx</i> handler use the CMSPHysicalFilesCacheMinutes web.config key (the default value is one week) and directly requested files use the IIS client cache configuration.</p> <p>See also: Caching files and resources</p>
Allow client cache revalidation	<p>If enabled, the server adds the must-revalidate directive to the Cache-control response header of file requests. Ensures that caching clients always perform revalidation when requesting expired files (most clients revalidate by default even without the <i>must-revalidate</i> directive).</p>
Output caching	
Enable output caching	<p>If enabled, the system allows output caching. Output cache stores the full HTML source of pages. If disabled, output caching is off on the whole website.</p> <p>To enable output caching for pages, configure the Output cache properties of individual pages in the Pages application on the Properties -> General tab. Both the main website setting and page settings must be enabled to use output caching.</p> <p>See also: Caching page output</p>
Cache output in file system (minutes)	<p>Specifies the number of minutes for which the system stores output cache in the file system. This provides persistent cache storage in case of application restarts.</p> <p>If not set, only the standard caching mechanism (in memory) is used. If you enter a value, the system loads the output data of requested pages from the file system cache when it is not found in the memory cache.</p> <p>To enable output caching in the file system for pages, configure the Allow file system cache property of individual pages in the Pages application on the Properties -> General tab.</p> <p>Warning: Enabling the file system output cache may cause your website to serve outdated content. When clearing invalid output cache due to changes made to pages or other dependencies, the system only removes cache files if the corresponding cache item also exists in the memory. The file system cache remains outdated if the memory cache is not available when the invalidation occurs (for example after an application restart or if you have different expiration times set for the two caching types).</p>



Output cache variables	<p>Determines under which conditions the system stores multiple versions of the output cache for pages. Click Edit to change the default settings.</p> <p>For example, the username variable ensures that the system stores a separate version of each page's output cache for every logged in user. If disabled, the output cache does not distinguish between users — if a page's output is cached for one user, the system may load the same content for all other users (depending on the remaining output cache variables).</p> <p>You can enable or disable separate output cache for:</p> <ul style="list-style-type: none"> • username - every logged in user (public users share the same output cache). • sitename - every site in the system (leave checked unless your sites all have identical content on pages with the same path). • lang - each language version of pages. • browser - different types of web browsers. • cookielevel - the cookie level preferences of visitors. • deviceprofile - the device profiles detected for visitors. • domain - the website domain name (check to disable output cache sharing between different domain aliases of sites). • viewmode - the page view modes used by Kentico, such as <i>design</i>, <i>edit</i>, <i>preview</i> or <i>live site</i>. Currently, the system only uses caching for the <i>live site</i> view mode. <p>The output cache <i>always</i> creates separate cache items based on the:</p> <ul style="list-style-type: none"> • Page path (including the virtual directory and extension) • Protocol in the request URL
Enable partial caching	<p>If enabled, the system allows partial caching of web parts. The partial cache stores the HTML output of individual web part instances. If not checked, partial caching is disabled on the whole website for all web parts.</p> <p>By default, web parts do not use partial caching. You need to enable partial caching for individual web part instances using the Partial cache minutes property.</p> <p>See also: Caching portions of the page output</p>
Partial cache variables	<p>Determines under which conditions the system stores multiple versions of the partial cache for web parts. Applies globally to the partial cache of all web parts.</p> <p>Click Edit to change the default settings.</p> <p>For example, the lang variable ensures that each web part instance has a separate version of the partial cache for every language. If disabled, the partial cache does not distinguish between languages — if a web part's output is cached for one language, the system loads the same content for all languages (depending on the remaining partial cache variables).</p> <p>You can enable or disable separate partial cache for:</p> <ul style="list-style-type: none"> • username - every logged in user (public users always share the same partial cache). • sitename - every site in the system (leave checked unless your sites have identical content). • lang - each language version of pages. • browser - different types of web browsers. • viewmode - the page view modes used by Kentico, such as <i>design</i>, <i>edit</i>, <i>preview</i> or <i>live site</i>. Currently, the system only uses caching for the <i>live site</i> view mode.

Resources (global settings only)

Allow resource compression	<p>If enabled, the system compresses JavaScript and CSS stylesheet web resources before sending them to the client browser. This reduces the amount of data that must be loaded. Uncompressed versions remain available for browsers that cannot process compressed data.</p> <p>Minification of the given resource type must be enabled (via the Allow JavaScript/CSS minification settings) in order for compression to be applied.</p>
Allow JavaScript minification	<p>Indicates if JavaScript resources should be minified before they are served to the client browser. Minified code has a reduced size, which saves bandwidth and decreases response times, but may not be suitable for debugging.</p> <p>Note: JavaScript minification does not work on instances running on Microsoft Azure Cloud Services.</p>
Allow CSS minification	<p>Indicates if CSS stylesheet resources should be minified before they are served to the client browser.</p>
CSS Styles (global settings only)	
Resolve macros in CSS	<p>If checked, macro expressions can be used in CSS stylesheets to dynamically insert the content of other stylesheets.</p> <p>To insert this type of macro, add an expression into the code of a stylesheet according to the following format: <code>{%CSS["<stylesheetname>"]%}</code></p> <p>Warning: If you wish to disable this setting, you first need to remove all occurrences of macros from your stylesheets. Unresolved macro expressions are not valid CSS code, so browsers cannot process stylesheets containing macros.</p>
Allow CSS from components	<p>If enabled, CSS styles defined for individual page components (e.g. web parts, page templates etc.) are automatically requested by the page where they are placed. Otherwise, you either need to have all styles defined in the website's stylesheet, or link the styles of the required components into the stylesheets via CSS macros.</p>
Combine CSS from components	<p>If enabled, pages load the CSS styles of all contained components via a single request. Otherwise different types of components each generate a separate request. The styles of multiple components of the same type (e.g. several web parts) are always retrieved by a single request.</p> <p>Combining the CSS requests into one may improve the load time of individual pages and is recommended in most cases.</p>

Related pages

- [Optimizing website performance](#)
- [Configuring caching](#)
- [Using code minification and compression](#)
- [Designing websites using CSS](#)