

Output caching is the most effective way to increase page performance. The output cache stores the full source code of pages, i.e. the HTML and client script that the server sends to browsers for rendering. When a visitor views a page, the server caches the output code in the application's memory. Until the cache expires, the system displays the page using the cached output. As a result, the application can quickly serve requests without running the page code or communicating with SQL servers.

The implementation is similar to native the ASP.NET output cache, with better performance and options. Also, the Kentico output cache has better support for clearing the cache than [external caching approaches](#).

***We recommend using output caching for all possible pages, especially when there is a chance of high load on individual pages.*** The drawback is that cached output can become outdated on dynamic pages, or if the website's content is updated. Output caching is not suitable for pages that need to refresh frequently (for example pages that update based on user input, or have randomly generated content). You cannot disable output caching for parts of pages.

However, the system provides several solutions that allow you to cache output even for pages with dynamic content:

- [Set dependencies](#) for the output cache (allows automatic clearing of the cache)
- Adjust how the system [caches multiple versions of pages](#)
- Use [substitution macros](#) to create dynamic content inside cached pages
- Disable full page output caching, but use [partial caching](#) for sections of the page (individual web parts)

## Configuring full page output caching

To use output caching for your website's pages:

### **Allow output caching globally:**

1. Open the **Settings** application.
2. Select the **System -> Performance** category.
3. Check **Enable output caching**.
4. **Save** the settings.

### **Enable output caching for individual pages:**

1. Open the **Pages** application.
2. Select the page in the content tree.
3. Open the **Properties -> General** tab.
4. Set the **Use output cache** property to **Yes** (in the **Output cache** section).
5. Specify the number of **Cache minutes**. The value must be greater than 0, and determines the expiration time of the page's output cache.

#### **Recommendations:**

- When used as the top-most level of caching (no [external caching](#)) - 1 minute to 1 hour.
  - When used with external caching - 10 minutes to 1 hour. Generally, 10 times the value of the external cache.
  - For almost completely static sections of the site - up to 1 day.
6. **Save** the page.



To enable output caching for all pages on your site, edit the properties of the root page. The underlying pages need to have the **Use output cache** property set to **Inherit** (this is the default state).

To apply different intervals to different sections - inherit the **Use output cache** property from different parent pages in the specific sections of the site. This allows you to set longer intervals for site sections with more static content.

When a visitor requests the page, the system saves the full output code into the cache. Until the cache expires, the system displays the cached output without processing the page.

## Setting output cache dependencies

The output cache of pages can become outdated when users update the website's pages and objects. Dependencies allow the system to automatically delete cached content whenever related objects are modified. The output cache provides **default dependencies** that clear the cache for pages:

- When a user modifies the given page's content, properties or [page template](#)
- Whenever a postback occurs on the page

By default, pages do NOT delete their output cache when you modify other pages or objects whose data is displayed on the page. For example, if a page contains a list of news articles and you update one of the news pages, the page displays the old content until the output cache expires.

To ensure that pages are always up-to-date, you need to set your own output cache dependencies:

1. Edit the page in the **Pages** application.
2. Switch to the **Design** tab.
3. Add the **Output cache dependencies** [web part](#) anywhere on the page.
4. Define the dependencies in the web part's **Cache dependencies** property. For more information and examples, refer to [Setting cache dependencies](#).
5. Click **OK**.

The system clears the page's output cache whenever an object specified by the dependencies is modified. With the cache cleared, the system fully processes the page on the next request, and caches the new output.

**Note:** If the page uses a shared [page template](#), the dependencies apply to all pages with the given template.

## Using persistent storage for the output cache

By default, the application stores the output cache of pages only in the memory. Websites lose all cached content if the application restarts. You can configure the system to save the output cache in the server's file system, which provides more persistent storage. The output cache then stores the content of the selected pages in both the application's memory and file system. The file system cache is used if the output of a requested page is not found in the memory cache.

Use persistent caching **together with regular** [output caching](#) to reduce memory footprint for sites that contain a large amount of content that you need to serve fast at any time.

**We recommend** enabling persistent storage if the application recycles often, or there is a possibility that it recycles during peaks.



**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 can only remove 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).

As a workaround, you can manually clear the file system output cache by clicking **Clear cache** on the **General** tab of the **System** application.

### To allow storage of output cache in the file system:

1. Go to **Settings -> System -> Performance**.
2. Type a number of minutes into the **Cache output in file system (minutes)** setting. The value must be greater than 0, and determines the expiration time of the output cache files.  
We recommend setting the interval to a value between 1 day (1440 minutes) and 1 week (10080 minutes).  
**Example:** if your website contains many pages and is under heavy load, you can reduce the memory footprint and still keep the cache available by configuring the interval to 1 day and configuring [output cache](#) to 10 minutes.
3. **Save** the settings.

### To enable persistent output caching for pages:

1. Open the **Pages** application.
2. Edit individual pages on the **Properties -> General** tab.
3. Set the **Allow file system cache** property to **Yes** (in the **Output cache** section).

4. **Save** the page.



To enable file system output caching for all pages on your site, edit the properties of the root page. The underlying pages need to have the **Allow file system cache** property set to **Inherit** (this is the default state).

## Deleting the output cache of pages

To manually clear the output cache of a page:

1. Open the **Pages** application.
2. Select the page in the content tree.
3. Open the **Properties** -> **General** tab.
4. Click **Clear output cache** in the **Output cache** section.

The page's output is removed from the cache in the server's memory. The next time a visitor opens the page, the system runs the page's code and saves the output into the cache again.

**Note:** The *Clear output cache* action does not delete the file system output cache. To manually clear the file system output cache, you need to delete all data cached by the application – open the **System** application and click **Clear cache** on the **General** tab.

## Caching portions of the page output

If you cannot use full output caching on a page, consider enabling **Partial caching** for sections of the page. The partial cache stores the HTML output of individual [web parts](#) (components that make up the content of pages). Partial caching allows the system to refresh pages on every request, but still improves performance by caching at least portions of the page output. You can set up partial caching for any number of web parts on a single page.

**Note:** Partial caching is not available on [precompiled websites](#).

Web parts do not use partial caching by default. You need to allow partial caching globally for your website, and then enable it for individual web part instances.

## Using partial caching

Use partial caching:

- When individual parts of the page need different cache intervals or segments of the page have rich output that cannot be easily handled by [output substitutions](#).
- If you have web parts that contain a lot of presentation code (for example, with nested API calls in transformations) then use partial caching over [content caching](#).
- For most of the master page content and any other content that is considered mainly static but is on pages where you cannot use full [output cache](#).

### To allow partial caching:

1. Open the **Settings** application.
2. Select the **System** -> **Performance** category.
3. Check the **Enable partial caching** setting.
4. **Save** the settings.

### To enable partial caching for web part instances:

1. Open the **Pages** application.
2. Edit the page containing the web part on the **Design** tab.
3. **Configure** the web part instance (double-click).
4. Expand the **Performance** category, and type a number of minutes into the **Partial cache minutes** property. The value must be greater than 0, and determines the expiration time of the web part's partial cache.

**Recommended settings:**

- Content changes often - 1 to 10 minutes.
- Mostly static content - 60 minutes to 1 day (1440 minutes).
- In combination with [output caching](#) on the same page, the interval should be about 10 times the interval set for the output cache.

5. Click **OK**.

When a visitor opens the page, the system saves the HTML output generated by the web part into the cache. Until the cache expires, the system renders the page without processing the web part, and displays the cached output instead.



#### Clearing of partial cache

By default, the system deletes the partial cache of web part instances:

- When you modify the property configuration of the given web part instance.
- Whenever a postback occurs on the page containing the web part.
- Editable web parts (*Editable text*, *Editable image*) automatically clear their partial cache when the content of the parent page is updated.
- Partial cache also clears whenever the application recycles and the cache doesn't support for [persistent storage](#).

We recommend that you [set additional cache dependencies](#) for web parts through the **Partial cache dependencies** property (in the **Performance** category). The system uses the dependencies to automatically clear the partial cache when the related objects are modified. Note that most web parts don't pick up the dependencies from their data sources, so you need to explicitly set the property.

To disable the default dependencies listed above, uncheck **Use default cache dependencies** and enable the **Preserve partial cache on postback** property.



**Tip:** If you wish to use partial caching on pages that are not based on the [portal engine](#), you can leverage standard [ASP.NET output caching for controls](#).

## Caching multiple output versions of pages

Web pages often display different content based on variable parameters, such as user permissions, language preferences, or the device and browser used to view the page. To compensate, the output cache stores multiple versions of pages. The system then automatically serves the appropriate cache version to visitors.



The output cache *always* creates separate cache items for pages according to the following variables:

- The page path (including the virtual directory and extension)
- The protocol in the request URL

You can configure the conditions that determine when the system uses separate or shared output cache:

1. Open the **Settings** application.
2. Select the **System -> Performance** category.
3. Click **Edit** next to the **Output cache variables** setting.
4. Enable or disable the following output cache variables:

Cache key variable	Ensures separate output cache for
username	Every logged in user. Public users share the same output cache.

sitename	Every site in the system.  <b>Important:</b> Leave the <i>sitename</i> variable checked unless your sites all have identical content on pages with the same path.
lang	Each <a href="#">language version</a> of pages.
browser	Different types of web browsers.
cookielevel	Different <a href="#">cookie level</a> preferences of visitors.
deviceprofile	Different <a href="#">device profiles</a> detected for visitors.
domain	Website domain names. Check to disable output cache sharing between <a href="#">domain aliases</a> 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.

##### 5. Click **Save**.

The system adds the selected variables and their values into the names of output cache keys. As a result, the cache keys are unique for all combinations of the required conditions.

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 you disable the **username** option, 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).



**Tip:** You can configure how the system shares [partial output cache](#) through the **Partial cache variables** setting. The variables apply globally to the partial cache of all web parts.

## Caching of personalized content

[Personalized content](#) (for example content contained in web part zones or personalized web parts) is cached per [personalization variant](#).

For example, when a page containing a web part zone with multiple personalization variants is served, the system caches only the personalization variant fitting the current request. When another request, meeting different personalization criteria, is made, the system reuses cached data for the static segments of the page (if still available) and caches only the different personalization variant of the web part zone. This way, the system avoids storing the static segments of the page multiple times.

## Using external caching approaches

You can also configure external caching approaches for your web application, such as [IIS Output caching](#), Content delivery network or Proxy caching. These are completely separate from the caching mechanisms in Kentico and are **applicable to any content** the application provides. When clients request a page whose valid output is cached on by the external cache, the server returns the response without requiring any processing from Kentico.

As these approaches do not support automatic cache clearing, their caching intervals define the largest amount of time for which the data on your may remain outdated.

### Recommendations:

- Configure very short intervals (30 seconds to 5 minutes) for sites experiencing very high peak loads.
- If your site mainly consists of static content, you can use longer intervals.
- Do not use floating intervals (intervals which extend when an item is accessed), as the cache isn't controlled by Kentico and therefore it could be hard to flush.



**Note:**

- External caches don't support [A/B testing](#), [MVT testing](#) or [content personalization](#), as they are usually very basic and can't tell one client context from another for the same URL.
- When using external static cache or output cache and your website uses postbacks, you should disable using of *ViewStateUserKey* to avoid ViewState inconsistencies introduced by output cached originally for other session.
- Kentico only supports the *User-mode* policy for IIS output caching. You cannot use Kernel-mode caching.