The performance of your website depends on many factors:

- The total number of pages on your website
- The complexity of the website (the depth of the content tree, number of page nesting levels, number of web parts per page)
- Custom functionality added to the application or individual sites
- The hardware and environment on which the Kentico application and database server are running
- The available resources if the server is shared with other applications

> ✅ **Tip: Finding sources of performance problems**
>
> - Check the system's event log for errors
> - Use the built-in debugging tools to investigate pages with low performance
> - Monitor your application using performance counters (EMS license required)

> ℹ️ **Optimizing on-line marketing performance**
>
> For high-traffic websites that use the Kentico EMS on-line marketing functionality, we strongly recommend following the Best practices for EMS performance in addition to the recommendations listed below.

Disable output filters

Applying output filters adds steps to the processing of pages, and may slow down the website. If the output code of your pages is valid without filtering, you can disable the filters:

1. Go to **Settings -> System -> Output filter**.
2. Choose your site in the **Site** selector.
3. Type the / character (forward slash) into the following settings:
   - **Excluded output form filter URLs**
   - **Excluded resolve filter URLs**
   - **Excluded XHTML filter URLs**
   - **Excluded HTML5 filter URLs**
4. **Save** the settings.

The system excludes all pages on the website from the output filter.

If you resolve your performance issues by disabling the output filter but need to keep the functionality to produce valid code, consider enabling output filter on web part level only. For example, you may want to enable output filter in web parts for which you cannot guarantee valid output, like the 'Editable text' web part.

Disable debugging tools

Always disable all Kentico debugging tools before deploying websites to the production environment:

1. Go to **Settings -> System -> Debug**.
2. Check the **Disable debugging** setting.
3. Click **Save**.

If you ever need to debug in the production environment, make sure you only enable the necessary Debug settings for a limited period of time.

Configure file storage and processing

Loading files from the database is a demanding operation. Storing files on the server's file system provides better performance in most cases:

Requirement: Your application needs to have the *Modify* permission for the web project directory in the server's file system.

1. Go to **Settings -> System -> Files**.
2. Enable the following settings:
    - **Store files in:** File system
    - **Generate thumbnails**
3. Consider disabling the following settings to improve loading time of *page attachments*:
    - **Check if files are published** (if you do not use <u>workflow or content scheduling</u> on your website)
    - **Check files permissions** (if you do need to restrict access to the attachments of <u>secured pages</u>)
4. If you don't need to protect your images with *watermarks*, we recommended disabling the watermarks feature to achieve the best performance. Watermarks don't generate thumbnails on the file system so the performance can be affected. Disable:
    - **Use watermark for page images**
    - **Use watermark for media files**
    - **Use watermark for object attachments**
5. Click **Save**.

See also: <u>Storing files</u>

Disable permission checks

If you don't need to check permissions for pages in the content tree, we recommended disabling the **Check page permissions** site setting to avoid unnecessary security checks. See <u>hiding pages based on permissions</u> for more information on the settings.

Organize your website's content in the most effective way

You can store your site's content either in *Pages*, Custom tables, *Module classes* or *Media libraries*. Each storage type is designed for a different type of data.

Storing large amounts of pages in the system can significantly affect the website's performance. Refer to <u>Defining website content structure</u> for more information. You can also reduce the amount of pages stored in the content tree by <u>setting up archival for outdated pages</u>.

Optimize how the system behaves when editing content

The following settings can significantly affect performance when creating and updating pages under a parent page that already has a large amount of descendant (nested in all levels) pages :

- **Settings -> Content -> Content management -> New page order** - we recommend using the 'Last' option in this settings as it is the least performance demanding.
- **Settings -> URLs and SEO -> Use name path for URL path** - automatically updates the page alias. Changes to page aliases for pages with a large amount of descendant pages can have huge performance impacts as the system has to update the paths for all the descendant pages as well.

Minimize lookup for content in the default culture

The following settings make the system look for default culture content if it isn't found in the requested (current) culture. This affects read performance.

Settings -> Content:

- **Combine with default culture** - to achieve better performance, consider translating pages that are often requested in non-default culture into additional cultures.
- **Combine files with default culture** - this setting is useful when the same *CMS.File* pages are used in all cultures. Replicating files into the most requested languages can positively affect the performance of individual requests.

Optimize your data source components (web parts and controls)

Check the configuration of all components (web parts and controls) that load data from the database or other sources. For best practices and examples, see Loading data efficiently.

Minimize compilation requirements for pages

If your pages have long initial load times (for example after application restarts), try to reduce the number of components that need to be compiled:

- Consider precompilation of the entire project before deploying to the production environment.
- Switch to the **HTML** type for page layouts that do not contain controls or inline code.
- Use the **Text / XML** type for simple transformations (if the code is pure HTML or only performs basic loading of field values). However, the *ASCX* transformation type generally provides better performance for complex transformations (with method calls or other advanced logic), particularly after the first request during which the transformation is compiled. Also, *ASCX* transformations have better performance on fully precompiled websites.

> ✅ **Tip - Optimizing dynamic compilation when developing web site projects**
>
> During the development of **web site** type projects, you may experience long compilation times and slow performance after making changes to top-level files (the *Global.asax* file and all files in the *Bin* and *App_Code* folders). By default, the application compiles the entire site after such changes, which can be a demanding operation.
>
> To optimize the dynamic compilation, set the **optimizeCompilations** attribute of the **<compilation>** element in your web.config file to *true*.
>
> ```
> <compilation debug="true" numRecompilesBeforeAppRestart="100" targetFramework="
> 4.6" optimizeCompilations="true">
> ```
>
> The dynamic compilation then only includes the files that were changed instead of the entire site.
>
> **Note**: Optimized dynamic compilation improves performance, but may cause compilation errors if you change an API or object that is used elsewhere on the site. For more information, see Understanding ASP.NET Dynamic Compilation.

Check custom code

If you integrated custom code into the application (including JavaScript), please make sure it works correctly.

- Avoid excessive database operations
- Reuse objects whenever possible
- Use custom caching when you load data through the API

Try to comment out your code and see if the performance improves.

Limit scheduled tasks

You can conserve system resources by limiting the execution of scheduled tasks:

- Disable all unnecessary tasks
- Increase the execution intervals of tasks
- Run tasks using the external Windows service if possible

Enable minification and compression of resources

Use code minification and compression to reduce the size of CSS and JavaScript resources. This helps limit the volume of traffic between the server and clients.

Configure caching for your website

Refer to:

- Configuring caching for detailed information about the caching mechanisms available in Kentico.
- Caching best practices for typical projects - for typical scenarios and our recommendations on how configure caching for best performance on these websites.

The following basic settings work well for most websites:

1. Go to **Settings -> System -> Performance**.
2. Choose your website in the **Site** selector.
3. Set the caching settings:
    - **Cache page info (minutes)**: 10 or more
    - **Cache content (minutes)**: 10 or more
    - **Cache files (minutes)**: 10 or more
    - **Client cache (minutes)**: 0
    - **Allow client cache revalidation**: yes (checked)
4. Click **Save**.

Note: Caching only improves performance on the second and subsequent page loads.

Use output caching

Output caching is the best way to significantly increase the performance of pages. We recommend caching page output whenever possible, particularly for high-traffic pages.

See Caching page output for detailed information.

To enable output caching:

1. Go to **Settings -> System -> Performance**.
2. Check **Enable output caching**.
3. **Save** the settings.
4. Set up output caching for individual pages in **Pages -> Edit -> Properties -> General -> Output cache**.

Note: You may need to take additional steps to ensure that cached pages do not display outdated content.

Configure Page info caching

Page info cache is a special content cache for URL rewriting. Configure page info caching according to the recommended intervals.

Increase the application pool Idle Time-out in IIS

If your website has periods of inactivity (no requests for over 20 minutes by default), users may experience long delays when first opening the site. To prevent this, set the **Idle time-out (minutes)** property of your IIS application pool to a higher value:

1. Open your **Internet Information Services (IIS) Manager** console (*Start -> Control Panel -> Administrative tools -> Internet Information Services (IIS) Manager*).
2. Select **<machine>/Application Pools**.
3. Right-click the application pool used by the Kentico application and select **Advanced Settings**.
4. Enter the required value into the **Idle Time-out (minutes)** property in the **Process Model** section.

Scale the hosting environment

If performance is still not satisfactory after you have taken all possible steps to optimize your website, you may need to upgrade your hosting environment:

- **Scale out** - add more servers hosting the same content, and set up a web farm
- **Scale up** - upgrade the hardware of your server

> ✅ One way to create an easily scalable website is to deploy your instance to cloud hosting.

*