

Many pages use components ([web parts](#) or controls) to load and display data from the Kentico database, or other external sources. Transferring data between storage spaces and the application can be a performance bottleneck.

To maximize data loading efficiency, consider the following points when setting up data source components.

Only load the data columns that you need

When loading data, you usually do not need all columns available in the source. The less data you retrieve, the faster your pages will be.

You can limit which columns data sources load through the **Columns** (or **Selected columns**) property. Enter a list of *column names* separated by commas. We recommend loading only the columns that your components need to display the required results:

- Columns used in the code of [transformations](#)
- Columns inside Where conditions, Order by clauses etc.

When loading Kentico pages, the behavior of the **Columns** property depends on the type of the data source:

Standard	<p>Standard page data sources load <i>all columns</i> if you leave the Columns property empty. The full list includes the columns of the View_CMS_Tree_Joined database view, the coupled data columns of individual page types, and the COM_SKU columns.</p> <p>By setting the Columns property, you specify exactly which of the columns the data source retrieves. When using web parts that retrieve data from multiple page types, you can specify only the columns common to all of the selected page types.</p> <p>Note: In addition to the columns that you specify in the Columns property, the data source always loads a set of specific columns that allow the system to identify individual pages (<i>NodeID</i>, <i>DocumentCulture</i>, etc.).</p>
Navigation	<p>Data sources of navigation components always load a set of default columns required to correctly display pages in menus.</p> <p>The Columns property only allows you to add extra columns. Leave the property empty unless your component requires a field that is not included in the defaults.</p> <p>Note: To find a full list of the default navigation columns, use the SQL queries debugging tool and inspect the query performed by your navigation component.</p>

To learn how Kentico stores pages in the database, refer to [Page database structure](#).

Load only the necessary page types

By default, a data source queries for all the available page types. You can improve the performance of the page by specifying only the page types you need. The most efficient use is to display one page type per web part whenever possible.

You can limit specify this limit using the **Page types** property.

Load data from sections with 100 000 pages at most

We recommend displaying data from content tree sections that contain 100 000 descendant pages at most. That is, if you're displaying data from an */Articles* section, make sure the section doesn't contain more than 100 000 pages in all levels. See [Defining website content structure](#) for more information on the limitations of storing large amounts of pages.

Disable unnecessary processing

[Page data sources](#) provide features for filtering data according to the settings and status of the website's pages. All such operations add steps to the data retrieval process and use the system's resources. By disabling filtering options that you do not need, you can reduce processing overhead when loading uncached data.

Consider the following when configuring the properties of page data sources:

- If you are not loading sensitive data ([secured pages](#)), disable the data source's **Check permissions** property.
- If the website's pages do not use [workflow or content scheduling](#) (i.e. pages are always published immediately), disable the **Select only published** property.
- If the web part often queries for pages that aren't translated in the current culture, consider providing a translation and disabling the **Combine with default culture** property. Disabling the property can significantly improve the page's performance.

Disable view state for web parts

By default, [web parts](#) store data in the page's [view state](#). If view state contains a large amount of information, it can slow down page performance.

You can turn off view state for web parts that do not need to preserve their state across postbacks:

1. Open the **Pages** application.
2. Select the page containing the web part.
3. On the **Design** tab, configure the given web part instance (double-click).
4. Enable the **Disable view state** property in the **Performance** category.

Disabling view state reduces processing overhead and the size of the page output that clients need to download. As a result, page load time improves (depending on the size of the view state).



Note: Test your web parts carefully after disabling view state. Web parts may not work correctly without view state in some scenarios.

Example - Optimizing a news list

The following example demonstrates how to maximize the performance of a **Repeater** web part. Repeaters load data from pages in the website's content tree, and use [transformations](#) to format the data into HTML output.

In this sample scenario, the Repeater displays news pages with the following configuration and environment:

- The website does not use [workflow](#), all pages are published.
- All of the website's pages are publicly available without security requirements.
- The Repeater web part:
 - loads **CMS.News** pages as the data
 - uses an **ORDER BY expression**: NewsReleaseDate DESC
 - renders the data using the following **Text** type transformations:

List item transformation

```
<div class="listBoxWithTeaser">
  <div class="teaser">
    {% GetImage(NewsTeaser, "", 90) %}
  </div>
  <div class="description">
    <a class="header bold" href="{% GetDocumentUrl() %}">
      {% HTMLEncode(NewsTitle) %}
    </a>
    <p>
      <span class="black bold">
        By {% GetUserFullName(NodeOwner) %}
      </span><br /><br />
      {% NewsSummary %}
    </p>
  </div>
</div>
```

Item detail transformation

```
<div class="listDetail">
  <h1>
    {% HTMLEncode(NewsTitle) %}
  </h1>
  <div class="teaser">
    {% GetImage(NewsTeaser, "", 230, 230, 500) %}
  </div>
  <div class="contentText contentTextTeaser">
    <div class="summary">
      <p>{% NewsSummary %}</p>
    </div>
    <div class="text">
      <p>{% NewsText %}</p>
    </div>
  </div>
</div>
```

To optimize how the Repeater loads data, perform the following steps:

1. Open the **Pages** application.
2. Configure the Repeater web part on the **Design** tab.
3. Set the following **Columns** (in the *Content filter* category):
[NewsReleaseDate](#), [NewsTeaser](#), [NewsTitle](#), [NewsSummary](#), [NewsText](#), [NodeOwner](#), [NodeAlias](#), [DocumentURLPath](#), [NodeAliasPath](#)



- Column used in the **ORDER BY** expression
- Columns accessed directly inside the transformations
- Column required by the **GetImage** method in the transformations
- Columns required by the **GetDocumentUrl()** transformation method

4. Disable **Select only published** (*Content filter* category).
5. Disable **Check permissions** (*System settings* category).
6. Check the **Disable view state** property (*Performance* category).
7. Configure [caching](#) for the web part:



- [Content caching](#) (caches the data loaded by the web part)
- [Partial output caching](#) (caches the web part's output code, the best option for static lists)

8. Click **OK**.

The web part now loads only the required data columns, skips unnecessary processing, and uses caching. This Repeater does not use filtering or paging, so it is safe to disable view state.