Storing a large amount of content in the content tree can have a significant impact on website performance. You can set up archival of outdated pages in your system so that you reduce the number of pages in the content tree.

We recommend the Archival section in the content tree approach for most projects, including projects with large amounts of pages (hundreds of thousands). For extremely content-heavy projects, consider using module classes or custom storage.

Archival section in the content tree

You can create a special section in the content tree into which you'll move outdated pages. An even better approach is to keep the archived pages in the same section and creating new sections for new content — structured by years, for example.

When using this approach, the listings you use to display published pages need be configured to only to load data only from specific sub-sections of the content tree. For example, in the following structure:

- *Articles*
    - *2015*
        - *Months*
            - *Articles1*
            - *Article 2*
            - ...
        - ...
    - *2014*
    - *2013*

Instead of displaying all the years in a single listing, individual web parts could be configured to cover only the pages stored under a specific year. You can configure this using the **Path** property.

Advantages

- You can easily display the archived pages on the live site.
- The archived pages can retain all their data, including the editing history.
- You can easily restore archived pages.

Disadvantages

- The archived pages are still part of the content tree. This approach can still have a performance impact.

Note: moving large sections of pages in the content tree is a performance demanding task. Make sure you don't perform the operation in peak hours.

Module classes

You can move the main data stored in the archived pages into module classes.

Advantages

- The archived pages are completely separated from the content tree.
- You can view the archived data via the administration UI.

Disadvantages

- To be able to store all the related objects, you have replicate their structure in module classes as well.
- You can only display the archived pages on the live site using a Repeater with custom query or by a your own implementation of a data source web part.
- Editing history is lost.
- The archived pages cannot be easily restored from the archive.

Custom storage

You can use the API to retrieve outdated pages and their related objects and store them in a custom storage. For example, you can serialize the data into an XML file.

Advantages

- The archived pages are completely separated from the content tree.
- The archived pages can retain all their data, including the editing history.
- You can restore the archived pages.

Disadvantages

- You cannot, by default, view the archived data via the administration UI.
- You cannot easily display the archived pages on the live site.