

The [staging](#) feature supports synchronization of the following types of files:

- [Page attachments](#)
- [Files in media libraries](#)

Note

The system only logs staging tasks for files when the changes are performed via the Kentico administration interface. If you make changes directly in the file system, for example upload or update files using FTP, the changes are not logged.

If you make a change to a file via the administration interface and then update the file using FTP, the current file (the one updated using FTP) will be transferred to the target server, even if the staging task was created before the file upload. This happens because the system loads the binary data of the files at the time of synchronization, not when the synchronization task is logged.

Compared with most objects in Kentico that support staging, files are often associated with large volumes of data. Processing and synchronizing the binary data of files can use up a considerable amount of memory on the servers and other network resources.

The system allows you to set limits on the size of files that are synchronized by staging. You can utilize such limits in the following typical scenarios:

Scenario	Recommendation
You need to stage files that are larger than the default limits (28.6 MB).	Set the maximum request and content length to the required values. Increase the limits to the lowest possible value that you need for your scenario. See: Setting the maximum content and request size
You wish to utilize staging in general, but restrict large files to avoid excessive resource consumption.	Limit the maximum content and request size. See: Setting the maximum content and request size If utilizing media libraries, limit the maximum size of staged media library files. See: Limiting staging of media library files
You need to synchronize very large files (hundreds of MB) between staging servers, and are running into errors (out of memory exceptions) when performing file operations with staging enabled.	Use the approach described in Synchronizing very large files .

Setting the maximum content and request size

Staging of files is restricted by the web application's maximum request limits:

- [Maximum request size](#) – the default .NET limit is 4096 kB (approximately 4 MB), but the default Kentico web.config contains a global `httpRuntime` element with a much higher value that will usually not be exceeded.
- [Maximum length of content in requests](#) – the default limit is approximately 28.6 MB.

To adjust these limits for the staging service:

1. On the **target server**, edit the **web.config** file in the **CMS** folder of the web project.
2. Add a **<location>** element for the staging service into the **<configuration>** section and set the following attributes:
 - **maxRequestLength** (`<system.web>/<httpRuntime>` element). The value is in **kiloBytes**.
 - **maxAllowedContentLength** (`<system.webServer>/<security>/<requestFiltering>/<requestLimits>` element). The value is in **Bytes**.

```
<location path="CMSPages/Staging">
  <system.web>
    <httpRuntime executionTimeout="2400" maxRequestLength="4096" />
  </system.web>
  <system.webServer>
    <security>
      <requestFiltering>
        <requestLimits maxAllowedContentLength="4096648" />
      </requestFiltering>
    </security>
  </system.webServer>
</location>
```



Note: The default web.config also contains a **<httpRuntime>** and **<requestLimits>** element under a location element that only affects WebDAV configuration. These default tags do NOT affect the request limits for staging or the application in general.

3. Save the web.config file.

Synchronizing files larger than the set value results in failure (the target server returns an *HTTP status 404: Not Found* error).



Important: The binary data of files only forms a part of the synchronization requests. You may need to set a higher maximum value than the file size that you wish to allow.

Limiting staging of media library files

To limit the maximum size of files staged from [media libraries](#):

1. On the **source server**, edit the **web.config** file in the **CMS** folder of the web project.
2. Add the **CMSMediaFileMaxStagingSize** key into the **<appSettings>** section of the web.config. The value is in **kiloBytes**.

```
<add key="CMSMediaFileMaxStagingSize" value="102400" />
```

3. Save the web.config file.

The system does not synchronize binary data for media library files above the limit. Synchronization tasks for such files only transfer the database definitions of the media objects, without the actual files. You can find records of files not being synchronized in the [Event log](#).

Synchronizing very large files

If you have staging enabled and work with extremely large files (typically around hundreds of MB, depending on the performance of your servers), you may encounter errors such as out of memory exceptions. The errors occur when the system attempts to process large volumes of binary data while preparing staging tasks for the file changes.

If you need to keep such large files synchronized between the servers in your environment, we recommend the following approach:

1. Store the files in a [media library](#) (i.e. not as page attachments).
2. Set up shared file system storage for the media library folder, and ensure that the storage location is accessible for all staging servers.
 - See [Configuring custom storage for media libraries](#).
 - You can also use a [custom file system provider](#), and map the media folder to a shared network location, [Azure Blob Storage](#) or [Amazon S3 storage](#).



3. Limit the size of binary data that is synchronized as part of media library file staging tasks – refer to the [Limiting staging of media library files](#) section.

Staging tasks related to large files now only synchronize the database definitions of the media files. The large binary data is no longer processed or transferred, but is automatically available for all servers due to the shared file system.



Important: Because of the shared file system, any changes to the binary data of the media files immediately apply to all staging instances. For example, if you update a video file with new content on a source staging server, the changes immediately apply to target staging servers, even before you synchronize the corresponding staging tasks containing related database data.