

Indexing tasks keep smart search indexes up-to-date according to changes made to content (for both [Azure Search indexes](#) and [locally stored indexes](#)). The system automatically creates indexing tasks when:

- A page or object covered by a search index is created, removed or has one of its fields modified
- An administrator **Rebuilds** or **Optimizes** a search index

To view a list of search indexing tasks that are waiting to be processed, open the **Smart search** application and select the **Azure tasks** or **Local tasks** tab.

Azure indexes

Azure tasks

Local indexes

Local tasks

Process tasks

Refresh

<div>≡</div> Actions	Task type	Object type	Additional data	Task created	Result
<div>🗑</div>	Update	Page	6;4	5/10/2017 2:05:52 PM	<a href="#">Failed</a>
<div>🗑</div>	Process	Content category	1	5/10/2017 2:34:49 PM	
<div>🗑</div>	Update	Page	14;8	5/10/2017 2:35:11 PM	

Items per page

25

By default, the system processes indexing tasks immediately when they are created. **The list of indexing tasks is usually empty** unless there is a long-running rebuild task, or the application is generating a very large number of indexing tasks.

If you encounter problems with new content not being indexed correctly, you can use the list of indexing tasks to investigate the issue. The list provides the following information about indexing tasks (depending on the task type):

- **Task type** – the action that the task performs with the data of the search index (Update, Delete, Rebuild, Optimize, Process)
- **Object type** – the type of object for which the task was created
- **Search field** – the index field used to find the correct data item inside the search index (**\_id** in most cases)
- **Task value / Additional data** – additional data used by the indexing task, typically an identifier of the related object (ID or code name)
- **Related object** – the name of the specific object related to the indexing task
- **Task created** – the date and time when the system created the indexing task
- **Result** – if an error occurs while processing the indexing task, you can view the error message details here

To clear the indexing queue or stop unnecessary indexing, **Delete** () individual tasks.



#### Dealing with task errors

If an indexing task results in an error, the system stops processing subsequent search tasks of the given type (Azure or local). Once you resolve the error or delete the problematic task, you can run any remaining search tasks by clicking **Process tasks** above the list.

## Disabling automatic processing for local indexing tasks

For [locally stored search indexes](#), you can disable automatic running of indexing tasks by adding the **CMSProcessSearchTasksByScheduler** key to the `/configuration/appSettings` section of your application's web.config file:

```
<add key="CMSProcessSearchTasksByScheduler" value="true" />
```

With the key set to true, the system creates local search indexing tasks, but does not run them. In this case, you need to regularly process the tasks using the **Execute local search tasks** [scheduled task](#). You cannot manually **Process tasks** when the key is enabled.



#### Executing local search tasks via the scheduler on a web farm

If your environment runs in a web farm *without a shared file system*, you need to configure the **Execute local search tasks** scheduled task to be executed on each of the web farm servers individually.

See: [Configuring scheduled tasks to be executed on every server of a web farm](#)



#### Processing of Azure Search indexing tasks

You cannot disable automatic processing for [Azure Search](#) indexing tasks. The processing is ensured by the **Execute Azure search tasks** [scheduled task](#). The task's scheduling interval is not important – the system automatically runs the task whenever a new Azure Search indexing task is logged (if another instance of the given scheduled task is not already running).