

Kentico provides a customization model that allows developers to extend or adjust how the system builds and maintains [Azure Search](#) indexes. Customization provides a way to set up Azure Search features that require advanced configuration, as well as adapt to changes in the Azure Search functionality and leverage new features.

For example, you can achieve the following scenarios by customizing the Azure Search:

- [Add scoring profiles for indexes](#)
- [Add suggesters for indexes](#)
- [Set language analyzers for index fields](#)
- [Register custom data types for Azure Search](#)
- [Configure the retry policy and batch size used by indexing operations](#)

To perform Azure Search customizations, you need to run code during the initialization of the application. Add the required code by creating a custom module class:

1. Open your Kentico solution in Visual Studio.
2. Create a new *Class Library* project in the Kentico solution (or reuse an existing custom project).
3. Add references to the required Kentico libraries (DLLs) for the new project:
  - a. Right-click the project and select **Add -> Reference**.
  - b. Switch to the **Browse** tab, click **Browse**, and navigate to the **Lib** folder of your Kentico web project.
  - c. Add references to the following libraries (and any others required for your code):

- **CMS.Base.dll**
- **CMS.Core.dll**
- **CMS.DataEngine.dll**
- **CMS.Search.Azure.dll**

4. Reference the custom project from the Kentico web project (*CMSApp* or *CMS*).
5. Edit the project's **AssemblyInfo.cs** file (in the *Properties* folder).
6. Add the **AssemblyDiscoverable** assembly attribute:

```
using CMS;  
  
[assembly:AssemblyDiscoverable]
```

7. Create the [custom module class](#) under your project.

You can now add Azure Search customizations by overriding the module's **OnInit** method. For most typical scenarios, you need to assign handler methods for [Azure Search indexing events](#).



#### Azure Search SDK requirement

Most types of customizations additionally require usage of the [Azure Search .NET SDK](#), which you can integrate by installing the **Microsoft.Azure.Search** [NuGet package](#) into your custom project.

## Configuring the Azure Search retry policy and batch size

In addition to custom handling of [Azure Search indexing events](#), the Kentico API allows developers to configure the retry policy and batch size that the system uses when performing Azure Search indexing operations.

Get an **instance** of the **SearchEngineConfiguration** class within the initialization code of your custom module class and set its properties (see the sections below).

### Retry policy

Because Azure Search indexes are cloud-based and hosted outside of the Kentico application, requests that interact with indexes may fail due to network errors, service unavailability, etc. To deal with such issues, the system sends all indexing requests using a retry policy with exponential backoff. Requests that fail are automatically repeated several times with exponentially growing time intervals.

If necessary, you can adjust the parameters of the retry policy by setting the following properties of the **SearchEngineConfiguration** class:

- **RetryCount** – sets the maximum number of retry attempts that the system performs after an indexing operation fails. Setting the value to 0 disables the retry functionality (not recommended). The default value is 5.
- **MaxBackoffTimeSeconds** – sets the ceiling for the time interval between retry attempts (in seconds). The default value is 8.

For example, if you set **RetryCount** to 7 and **MaxBackoffTimeSeconds** to 16, the system retries failed indexing operations up to 7 times, with the following intervals between the attempts (in seconds): 0, 1, 2, 4, 8, 16, 16

## Batch size

The **DocumentsBatchSize** property of the **SearchEngineConfiguration** class sets the maximum number of [search documents](#) processed by a single request to the Azure Search service. For example, when building a new index that covers 2500 pages in Kentico with a maximum batch size of 1000, the system first creates the Azure Search index with 1000 documents, and then updates the index twice by adding 1000 and 500 documents.

The default document batch size value is **1000**. When setting a custom value, you need to respect the [API Request limits of the Azure Search service](#). For example, you may need to decrease the batch size if the total request size exceeds the maximum limit of 16 MB (when indexing a large number of objects with long text values).



### Azure Search document batch size vs. Index batch size

The maximum number of search documents processed per request is also limited by the **Batch size** setting of individual search indexes, which can be configured in the Kentico administration interface. The index batch size sets the maximum number of records that the system retrieves per query when loading data from the Kentico database, which effectively also limits the document batch size of Azure Search requests. By default, the index batch size is **500**.

The **SearchEngineConfiguration.DocumentsBatchSize** property only has an effect for indexes whose batch size setting is greater than the property's value.

## Example

```
using CMS;
using CMS.DataEngine;
using CMS.Search.Azure;

// Registers the custom module into the system
[assembly: RegisterModule(typeof(CustomAzureSearchModule))]

public class CustomAzureSearchModule : Module
{
    // Module class constructor, the system registers the module under the name
    "CustomAzureSearch"
    public CustomAzureSearchModule()
        : base("CustomAzureSearch")
    {
    }

    // Contains initialization code that is executed when the application starts
    protected override void OnInit()
    {
        base.OnInit();

        var azureSearchConfiguration = SearchEngineConfiguration.Instance;

        // Customizes the Azure Search retry policy
        azureSearchConfiguration.RetryCount = 7;
        azureSearchConfiguration.MaxBackoffTimeSeconds = 16;

        // Sets the maximum number of search documents processed by each request to
        the Azure Search service
        azureSearchConfiguration.DocumentsBatchSize = 500;
    }
}
```

## Reference - Azure Search events

This section provides an overview of system events that developers can handle to customize how the system builds and maintains Azure Search indexes.



For general information about Azure Search indexes and their structure, refer to the [Create an Azure Search index](#) article.

## Indexes

Class: **SearchServiceManager**

Event	Event types	Description
-------	-------------	-------------

CreatingOrUpdatingIndex	Execute	<p>Occurs when the system sends requests to create or update the definition of an index within an Azure Search service, for example when rebuilding Azure Search indexes in Kentico.</p> <p>The event is NOT triggered when adding, updating or removing documents within an existing index, unless an update of the index's definition is required (e.g. when adding a document with new fields for the first time).</p> <p>Depending on the number of indexed pages or objects and the used batch size, the event may occur multiple times when building a single search index – separately for each batch of processed search documents that include a new field not yet contained by the index. When implementing handlers for the event, always consider cases where the related index already exists and does not yet contain all possible fields.</p> <p>Examples of use:</p> <ul style="list-style-type: none"> <li>• <a href="#">Adding a scoring profile for an index</a></li> <li>• <a href="#">Adding a suggester for an index</a></li> </ul> <p>Handler parameters: <b>CreateOrUpdateIndexEventArgs</b></p> <ul style="list-style-type: none"> <li>• <b>Index</b> (Microsoft.Azure.Search.Models.Index) – Azure Search .NET SDK object representing the related index.</li> <li>• <b>SearchService</b> (CMS.Search.Azure.SearchService) – provides information about the Azure Search service specified for the index.</li> </ul>
-------------------------	---------	--

## Documents

Class: **DocumentCreator** (access an **Instance** of the class to assign event handlers)

Event	Event types	Description
CreatingDocument	Before After Failure	<p>Occurs when the system creates individual documents within an Azure Search index. Triggered separately for every indexed object.</p> <p>Use the <b>Before</b> event if you wish to modify the processed source data for the given document (<i>SearchDocument</i> property of the handler's <i>CreateDocumentEventArgs</i> parameter). Use the <b>After</b> event to adjust the properties of the resulting <i>Document</i> object (<i>Document</i> property of the handler's <i>CreateDocumentEventArgs</i> parameter).</p> <p>Handler parameters: <b>CreateDocumentEventArgs</b></p> <ul style="list-style-type: none"> <li>• <b>Document</b> (Microsoft.Azure.Search.Models.Document) – Azure Search .NET SDK object representing the related document.</li> <li>• <b>SearchDocument</b> (CMS.DataEngine.SearchDocument) – object holding the fields and values of the indexed data for the given object.</li> <li>• <b>Searchable</b> (CMS.DataEngine.ISearchable) – the Kentico object whose data is being indexed. Can be converted to a specific <a href="#">Info object</a>.</li> <li>• <b>SearchIndex</b> (CMS.DataEngine.ISearchIndexInfo) – object representing the related search index in Kentico.</li> </ul>

Adding DocumentValue	Execute	<p>Occurs when the system sets the values of individual fields for documents within an Azure Search index. Triggered separately for each field of every indexed object.</p> <p>You can use the event to modify the <b>AzureName</b> and <b>Value</b> properties of the handler's <i>AddDocumentValueEventArgs</i> parameter before the value is converted to an <a href="#">Azure Search data type</a> and saved to the document.</p> <p>Handler parameters: <b>AddDocumentValueEventArgs</b></p> <ul style="list-style-type: none"> <li>• <b>AzureName</b> (string) – name of the resulting field in the Azure Search document (Microsoft.Azure.Search.Models.Document).</li> <li>• <b>Document</b> (Microsoft.Azure.Search.Models.Document) – Azure Search .NET SDK object representing the related document.</li> <li>• <b>Name</b> (string) – the name of the source field in the <i>SearchDocument</i> data.</li> <li>• <b>SearchDocument</b> (CMS.DataEngine.SearchDocument) – object holding the fields and values of the indexed data for the given object.</li> <li>• <b>Searchable</b> (CMS.DataEngine.ISearchable) – the Kentico object whose data is being indexed. Can be converted to a specific <a href="#">Info object</a>.</li> <li>• <b>SearchIndex</b> (CMS.DataEngine.ISearchIndexInfo) – object representing the related search index in Kentico.</li> <li>• <b>Value</b> (object) – the value assigned to the resulting field in the Azure Search document.</li> </ul>
----------------------	---------	--

## Fields

Class: **DocumentFieldCreator** (access an **Instance** of the class to assign event handlers)

Event	Event types	Description
CreatingField	Before After Failure	<p>Occurs when the system creates individual fields for a document within an Azure Search index. Triggered separately for each field of every indexed object.</p> <p>Use the <b>Before</b> event if you wish to manually initialize the <i>Field</i> object (Microsoft.Azure.Search.Models.Field). Use the <b>After</b> event to adjust the properties of the resulting <i>Field</i> object (<i>Field</i> property of the handler's <i>CreateFieldEventArgs</i> parameter).</p> <p>Example of use: <a href="#">Setting language analyzers for index fields</a></p> <p>Handler parameters: <b>CreateFieldEventArgs</b></p> <ul style="list-style-type: none"> <li>• <b>Field</b> (Microsoft.Azure.Search.Models.Field) – Azure Search .NET SDK object representing the related field.</li> <li>• <b>SearchField</b> (CMS.DataEngine.ISearchField) – object representing the search settings of the field in Kentico.</li> <li>• <b>Searchable</b> (CMS.DataEngine.ISearchable) – the Kentico object whose data is being indexed. Can be converted to a specific <a href="#">Info object</a>.</li> <li>• <b>SearchIndex</b> (CMS.DataEngine.ISearchIndexInfo) – object representing the related search index in Kentico.</li> </ul>

Creating Fields	Before After Failure	<p>Occurs when the system creates the set of fields for a document within an Azure Search index. Triggered separately for every indexed object.</p> <p>Use the <b>Before</b> event if you wish to modify or extend the collection of processed source fields (<i>SearchFields</i> property of the handler's <i>CreateFieldsEventArgs</i> parameter). Use the <b>After</b> event to modify the resulting list of Azure Search fields (<i>Fields</i> property of the handler's <i>CreateFieldsEventArgs</i> parameter).</p> <p>Handler parameters: <b>CreateFieldsEventArgs</b></p> <ul style="list-style-type: none"> <li>• <b>Fields</b> (List&lt;Microsoft.Azure.Search.Models.Field&gt;) – list of Azure Search .NET SDK objects representing the related fields.</li> <li>• <b>SearchFields</b> (IEnumerable&lt;CMS.DataEngine.ISearchField&gt;) – collection of objects representing the search settings of the given fields in Kentico.</li> <li>• <b>Searchable</b> (CMS.DataEngine.ISearchable) – the Kentico object whose data is being indexed. Can be converted to a specific <a href="#">Info object</a>.</li> <li>• <b>SearchIndex</b> (CMS.DataEngine.ISearchIndexInfo) – object representing the related search index in Kentico.</li> </ul>
-----------------	----------------------------	--



#### Kentico system fields

Search indexes in Kentico contain special system fields that are required for general functionality and various other features. For example `_content`, `_index`, `_idcolumnname`. We do not recommend modifying or renaming the system fields within custom handlers of field or search document events.

In most cases, the names of system fields in Kentico start with an underscore. When creating matching fields within Azure Search indexes, the **sys\_** prefix is used instead (to comply with the [Azure Search naming rules](#)).