Azure Search uses Language analyzers to process text values. By default, searchable fields are analyzed with the Apache Lucene Standard analyzer. Because Azure Search supports a variety of languages, it additionally provides other text analyzers with more advanced capabilities for specific languages.

If you wish to assign a specific analyzer to the fields of an index, customize the functionality that Kentico uses to build Azure Search indexes:

1. Open your Kentico solution in Visual Studio.
2. Create a custom module class.
3. Override the module's **OnInit** method and assign a handler to the **DocumentFieldCreator.Instance.CreatingField.After** event.
4. Perform the following in the event's handler method:
    a. Write a condition to limit which search indexes and fields are affected by the analyzer customization (using values of the **SearchIndex** and **SearchField** properties of the handler's *CreateFieldEventArgs* parameter).
    b. Access the **Field** property of the handler's *CreateFieldEventArgs* parameter, and assign a valid analyzer name into one of its *Analyzer* properties.

    > **Field analyzer properties**
    >
    > The **Field** class (Microsoft.Azure.Search.Models.Field) offers the following properties for setting the analyzer:
    >
    > - **Analyzer**
    > - **IndexAnalyzer** (used at indexing time)
    > - **SearchAnalyzer** (used at search time)
    >
    > For more information, see the **Index Attributes** section of the Create Index article.

5. Sign in to the Kentico administration interface.
6. Open the **Smart search** application and **Rebuild** any related Azure Search indexes.

The customized Azure Search index fields now use the specified language analyzer.

Example

The following example demonstrates how to set the language analyzer for the *skudescription* field of an Azure Search index named *dg-store*.

Start by preparing a separate project in your Kentico solution for the custom module class:

1. Open your Kentico solution in Visual Studio.
2. Create a new *Class Library* project in the Kentico solution named **SearchCustomization**.
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:
        - **CMS.Base.dll**
        - **CMS.Core.dll**
        - **CMS.DataEngine.dll**
        - **CMS.Search.Azure.dll**
4. Right-click the *SearchCustomization* project in the **Solution Explorer** and select **Manage NuGet Packages**.
5. Install the **Microsoft.Azure.Search** package.
6. Reference the *SearchCustomization* project from the Kentico web project *(CMSApp* or *CMS)*.
7. Edit the *SearchCustomization* project's **AssemblyInfo.cs** file (in the *Properties* folder).
8. Add the **AssemblyDiscoverable** assembly attribute:

```
using CMS;

[assembly:AssemblyDiscoverable]
```

Continue by implementing the custom module class and rebuilding the related search index:

1. Create a new class named **CustomAzureSearchModule** under the *SearchCustomization* project, with the following code:

```
using System;

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();

        // Assigns a handler to the CreatingField.After event for Azure Search
indexes
        DocumentFieldCreator.Instance.CreatingField.After +=
UseCustomSearchAnalyzer;
    }

    private void UseCustomSearchAnalyzer(object sender, CreateFieldEventArgs e)
    {
        string indexName = e.SearchIndex.IndexCodeName;
        string fieldName = e.SearchField.FieldName;

        // Sets the 'en.microsoft' analyzer for the 'skudescription' field in the
'dg-store' index
        if (indexName.Equals("dg-store", StringComparison.
InvariantCultureIgnoreCase)
            && fieldName.Equals("skudescription", StringComparison.
InvariantCultureIgnoreCase))
        {
            e.Field.Analyzer = "en.microsoft";
        }
    }
}
```

2. Save all changes and **Build** the *SearchCustomization* project.
3. Sign in to the Kentico administration interface.

4. Open the **Smart search** application and **Rebuild** the *dg-store* index.

The *skudescription* field of the *dg-store* index now uses the "en.microsoft" language analyzer, both during indexing and at search time.