

Azure Search index fields and the fields of objects in Kentico each use a different set of data types. See the [Supported data types \(Azure Search\)](#) article for details.

When building Azure Search indexes, Kentico performs data type mapping and conversions for field values. By default, the following data type mapping is used:

Kentico field data type	C# type	Azure Search data type
Boolean (Yes/No)	bool	Edm.Boolean
Date Date and time	DateTime	Edm.DateTimeOffset
Decimal number	decimal	Edm.Double
Floating point number (Double precision)	double	Edm.Double
Integer number	int	Edm.Int32
Long integer number	long	Edm.Int64
Long text Text	string	Edm.String
Time interval	TimeSpan	Not mapped
Unique identifier (GUID)	Guid	Edm.String

If you wish to add values with a custom data type into Azure Search indexes, you first need to register the given data type, and prepare a conversion function that maps the type to one of the types supported by Azure Search. You can also use the same approach if you wish to change how the default Kentico data types map to Azure Search types.

1. Open your Kentico solution in Visual Studio.
2. Create a [custom module class](#).
3. Override the module's **OnInit** method and call the **RegisterMapping** method of the **CMS.Search.Azure.DataMapper** class. The method's parameters specify:
 - The original C# data type
 - The corresponding value from the **Microsoft.Azure.Search.Models.DataType** enumeration
 - A conversion method
4. Implement the conversion methods used in your **RegisterMapping** calls.

When creating Azure Search indexes, the system now uses the specified data types for the appropriate fields.



Index rebuild required

To apply the custom data types to existing Azure Search indexes:

1. Sign in to the Kentico administration interface.
2. Open the **Smart search** application.
3. **Rebuild** any related Azure Search indexes.

Example

The following example demonstrates how to re-map the **Decimal number** data type of Kentico fields to the **Edm.String** data type used by Azure Search.

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 indexes:

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



```
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();

        // Registers custom mapping of the decimal data type in Kentico to
        strings in Azure Search indexes
        DataMapper.Instance.RegisterMapping(typeof(decimal), Microsoft.Azure.
        Search.Models.DataType.String, ConvertDecimalToString);
    }

    // Converts decimal values to strings
    // The resulting strings use ',' as the group separator, '.' as the decimal
    separator, and two decimal places
    private string ConvertDecimalToString(object decimalValue)
    {
        return ((decimal)decimalValue).ToString("N2", System.Globalization.
        CultureInfo.InvariantCulture);
    }
}
```

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** your Azure indexes.

Any Kentico object fields that use the *Decimal number* data type are now created as *Edm.String* type fields within Azure Search indexes.