Custom extractors allow you to use the attachment search feature of page indexes for non-default file types.

To define a custom search text extractor:

1. Create a class that implements the **ISearchTextExtractor** interface (**CMS.Search** namespace).
   - You can add the class into a custom assembly within the Kentico solution (recommended) or directly into the Kentico web project (into a custom folder under the **CMSApp** project for *web application* installations, into the **App_Code** folder for *web site* installations).
2. Define the **ExtractContent** method.

   The ExtractContent method has the following parameters:

   - **BinaryData data** - contains the data of the attachment file.
   - **ExtractionContext context** - provides information about the page whose attachment is being indexed. Use the *context.Culture* property to get the culture code of the page's language version.

   The method must return an **XmlData** object. The system adds the object's data to page search indexes for the given attachment. Use the **XmlData.SetValue** method to enter values into individual search index fields.
3. Register the extractor for a file extension by calling the **SearchTextExtractorManager.RegisterExtractor** method.

```
SearchTextExtractorManager.RegisterExtractor("extension", new
CustomExtractorClass());
```

   Note: You need to call the *RegisterExtractor* method on application start (when the application and modules are being initialized).
4. Rebuild your page search indexes.

The search uses the custom extractor to index page attachments with the specified extension.

Modifying custom extractors

The system stores the text content extracted from page attachments in the database. When rebuilding page indexes, the search loads the "cached" attachment text from the database. The text extraction process only runs for attachments that do not have any search content saved.

If you change the functionality of a custom extractor, you need to clear the attachment content from the database:

1. Sign in to the Kentico administration interface and open the **System** application.
2. Select the **Files -> Attachments** tab.
3. Click **Clear attachment search cache**.

You can then **Rebuild** your page indexes, which updates the attachment content according to your extractor's new functionality.

Example - Implementing a search text extractor

The following example demonstrates how to create a very basic extractor for .txt files:

1. Open your Kentico web project in Visual Studio (using the **WebSite.sln** or **WebApp.sln** file).
2. Create a new class, for example named **CustomSearchTextExtractor.cs**.

- Either add the class into a custom project within the Kentico solution (recommended) or directly into the Kentico web project (into a custom folder under the **CMSApp** project for *web application* installations, into the **A pp_Code** folder for *web site* installations).

3. Add the following *using* statements to the class:

```
using CMS.Search;
using CMS.Core;
using CMS.Helpers;
using CMS.IO;
using CMS.DataEngine;
```

4. Make the class implement the **ISearchTextExtractor** interface.

```
public class CustomSearchTextExtractor : ISearchTextExtractor
```

5. Define the **ExtractContent** method:

```
public XmlData ExtractContent(BinaryData data, ExtractionContext context)
{
        if (data.Stream != null)
        {
                // Reads the text file's data stream
                data.Stream.Position = 0;
                string resultString = StreamReader.New(data.Stream).ReadToEnd();

                // Adds text into the CONTENT search index field
                XmlData contentData = new XmlData();
                contentData.SetValue(SearchFieldsConstants.CONTENT, resultString);

                return contentData;
        }

        return null;
}
```

6. Save the **CustomSearchTextExtractor.cs** file.
7. To register your custom extractor, create a custom module class in the same location.

For basic execution of initialization code, you only need to register a "code-only" module through the API. You do NOT need to create a new module within the **Modules** application in the Kentico administration interface.

```
using CMS;
using CMS.Base;
using CMS.DataEngine;
using CMS.Search;

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

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

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

                // Registers the CustomSearchText extractor for the .txt extension
                SearchTextExtractorManager.RegisterExtractor("txt", new
CustomSearchTextExtractor());
        }
}
```

8. Save the module class.
9. Sign in to the Kentico administration interface.
10. Open the **Smart search** application and **Rebuild** your page search indexes.

The example is only intended as a demonstration – the system already contains a default extractor for .txt files. You can use the same approach to create extractors for other file types.