

# How to find text in PDF with smart match in C# with ByteScout PDF Extractor SDK

Write code in C# to find text in PDF with smart match with this step-by-step tutorial

The coding tutorials are designed to help you test the features without need to write your own code. ByteScout PDF Extractor SDK is the Software Development Kit (SDK) that is designed to help developers with data extraction from unstructured documents like pdf, tiff, scans, images, scanned and electronic forms. The library is powered by OCR, computer vision and AI to provide unique functionality like table detection, automatic table structure extraction, data restoration, data restructuring and reconstruction. Supports PDF, TIFF, PNG, JPG images as input and can output CSV, XML, JSON formatted data. Includes full set of utilities like pdf splitter, pdf merger, searchable pdf maker. It can find text in PDF with smart match in C#.

This code snippet below for ByteScout PDF Extractor SDK works best when you need to quickly find text in PDF with smart match in your C# application. In your C# project or application you may simply copy & paste the code and then run your app! Code testing will allow the function to be tested and work properly with your data.

Trial version of ByteScout PDF Extractor SDK can be downloaded for free from our website. It also includes source code samples for C# and other programming languages.

C# - Program.cs

```
using System;
using Bytescout.PDFExtractor;

namespace FindTextSmartMatch
{
    class Program
    {
        static void Main(string[] args)
        {
            TextExtractor extractor = new TextExtractor("demo", "demo");

            // Load the document
            extractor.LoadDocumentFromFile("sample2.pdf");

            // Smart match the search string like Adobe Reader
            extractor.WordMatchingMode = WordMatchingMode.SmartMatch;

            string searchString = "land";

            // Get page count
            int pageCount = extractor.GetPageCount();
        }
    }
}
```

```
// Iterate through pages
for (int i = 0; i < pageCount; i++)
{
    // Search for text string
    if (extractor.Find(i, searchString, false))
    {
        do
        {
            // Output search results
            Console.WriteLine("Found on page " + i + " at location " +
extractor.FoundText.Bounds.ToString());

            // Now we are getting the found text
            string extractedString = extractor.FoundText.Text;
            Console.WriteLine("Found text: " + extractedString);
        }
        while (extractor.FindNext()); // Search next occurrence of the
search string
    }
}

// Cleanup
extractor.Dispose();

Console.WriteLine();
Console.WriteLine("Press any key to exit...");
Console.ReadKey();
}
}
}
```

---

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout PDF Extractor SDK](#)

[Explore documentation](#)

[Visit www.ByteScout.com](#)

or

[Get Your Free API Key for www.PDF.co Web API](#)