Microsoft

May 10-11, 2023
sambaXP Conference

Microsoft Interoperability Track

Microsoft

Redact PII from a Word Document Using the Open XML SDK and Azure Cognitive Services for Language

Michael Bowen
Escalation Engineer - Microsoft Office Open Specifications

Agenda

- Introduction to Azure Cognitive Services
- Create App using the Open XML SDK and Azure Cognitive Services
- How the Office Open XML File Format Works

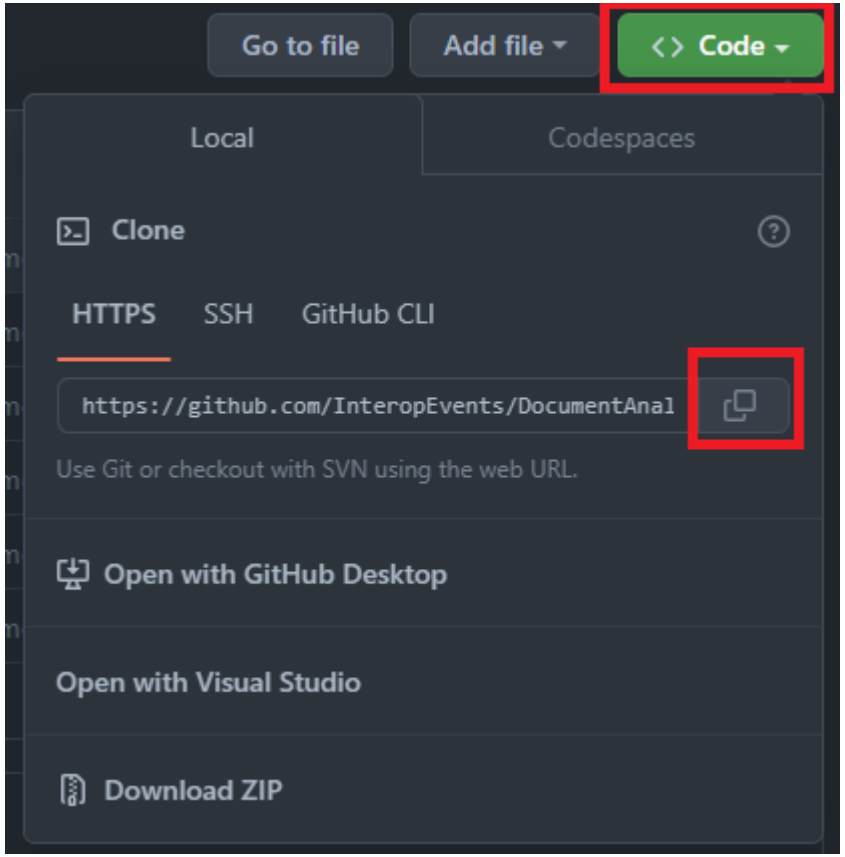
Prerequisites

- ✓ Visual Studio (free community edition is OK)
- ✓ .NET 6.0
- ✓ git command line tools
- ✓ Word Processing app that can open .docx files such as LibreOffice Writer or Microsoft Word
- ✓ Free Account with Azure

Project Code and Tutorial Are Available on GitHub

https://github.com/InteropEvents/DocumentAnalyzer

```
git clone https://github.com/InteropEvents/DocumentAnalyzer.git
```



Azure Cognitive Services

What are Azure Cognitive Services?

Azure Cognitive Services are cloud-based artificial intelligence (AI) services that help developers build cognitive intelligence into applications.

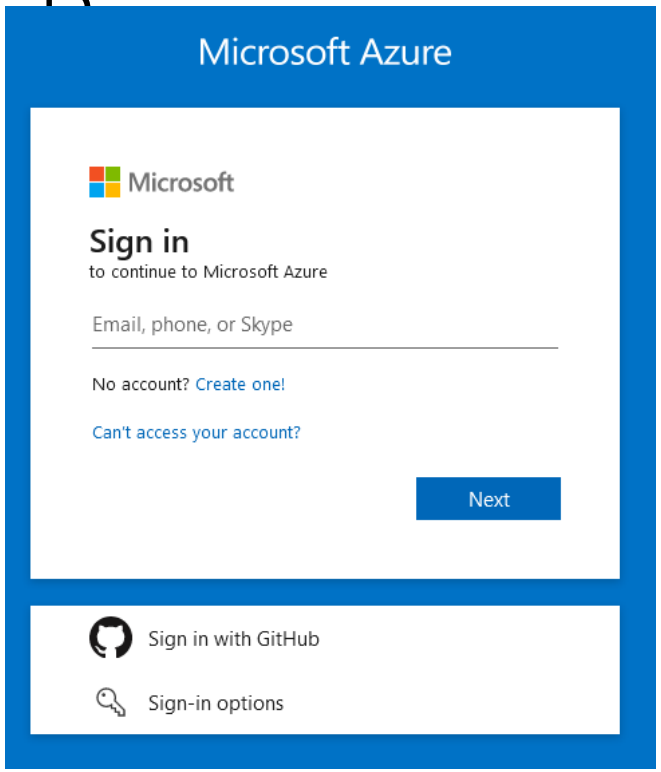
Four Main Categories of Cognitive Services

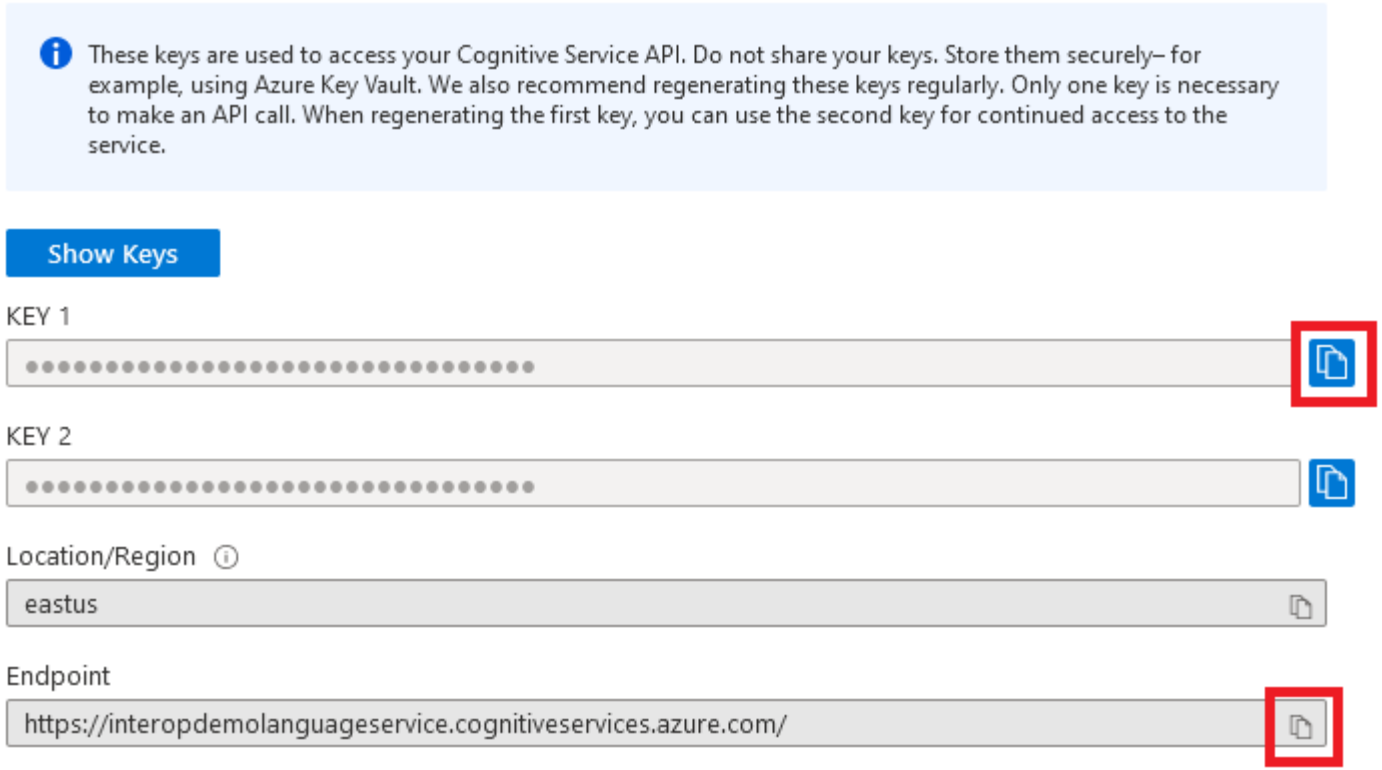
- Vision
- Speech
- Decision
- **Language**

Language Service

- Extract Key Phrases
- Find Linked Entities
- Named Entity Recognition (NER)
- Custom Named Entity Recognition (Custom NER)
- Text Analytics for Health
- **Personally Identifiable Information (PII) Detection**

Create Azure Cognitive Service for Language





Go to portal.azure.com

Sign in or create a new account

Create an Azure Cognitive Resource

Copy the key and endpoint

Complete instructions on https://github.com/InteropEvents/DocumentAnalyzer

Free Tier

Azure Cognitive Services for Language has a free tier, which allows 5,000 free text records per month.

Instance	Features	Inferencing	Training and model endpoint hosting
Free - Web	Sentiment analysis (available in containers) Key phrase extraction (available in containers) Language detection (available in containers) Custom question answering[1] Prebuilt question answering Named entity recognition, including PII	5,000 text records free per month	N/A
	Conversational language understanding		Standard training: free Advanced training: up to 1 hour free Model endpoint hosting: free
	Custom text classification Custom named entity recognition[2]		Training: up to 1 hour free Model endpoint hosting: up to 1 model free

Ways to Consume Azure Cognitive Services

C#

Azure.AI.TextAnalytics

Java

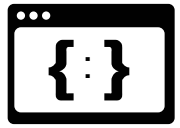
azure-ai-textanalytics

JavaScript

@azure/ai-text-analytics

Python

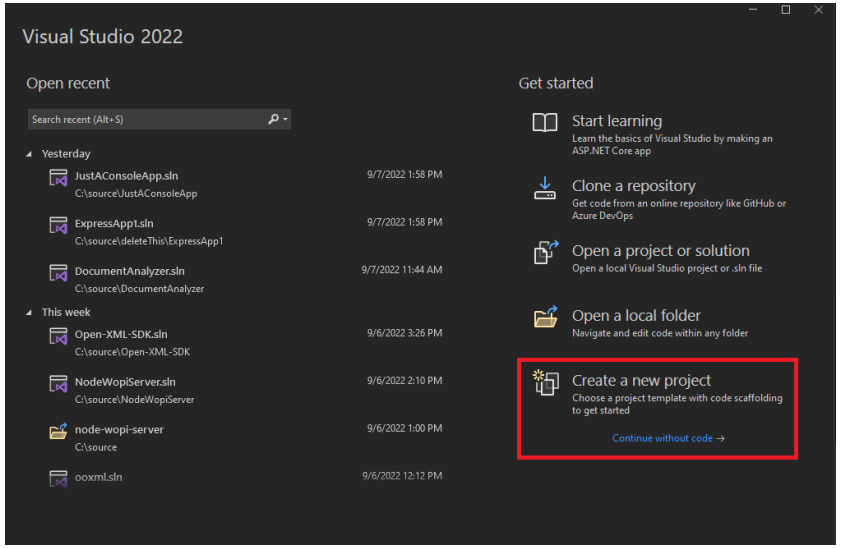
azure-ai-textanalytics



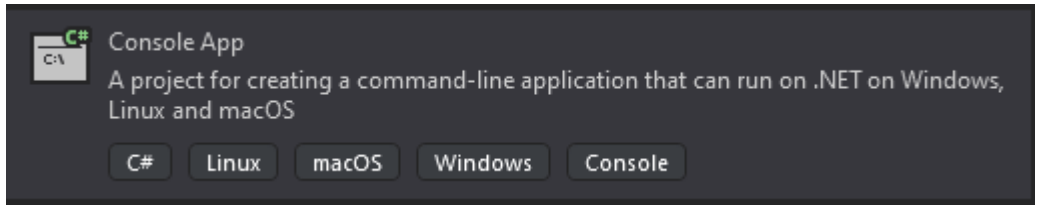
REST API

Creating the Application

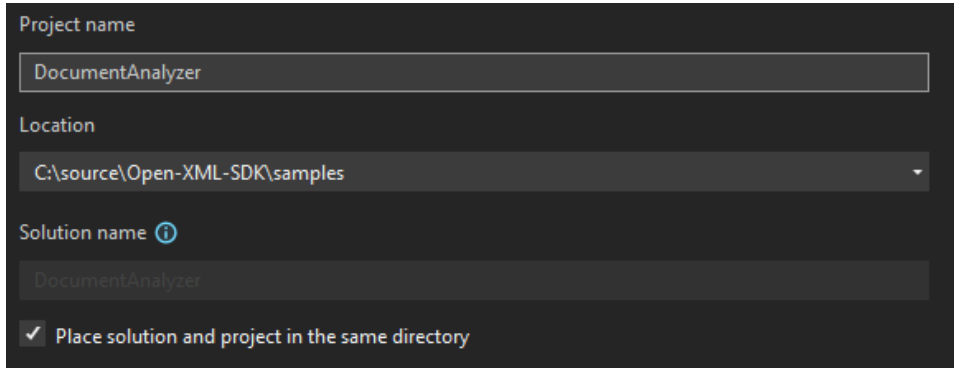
Create a Console Application with Visual Studio



Open Visual Studio and select create a new project



Select Console App and click Next

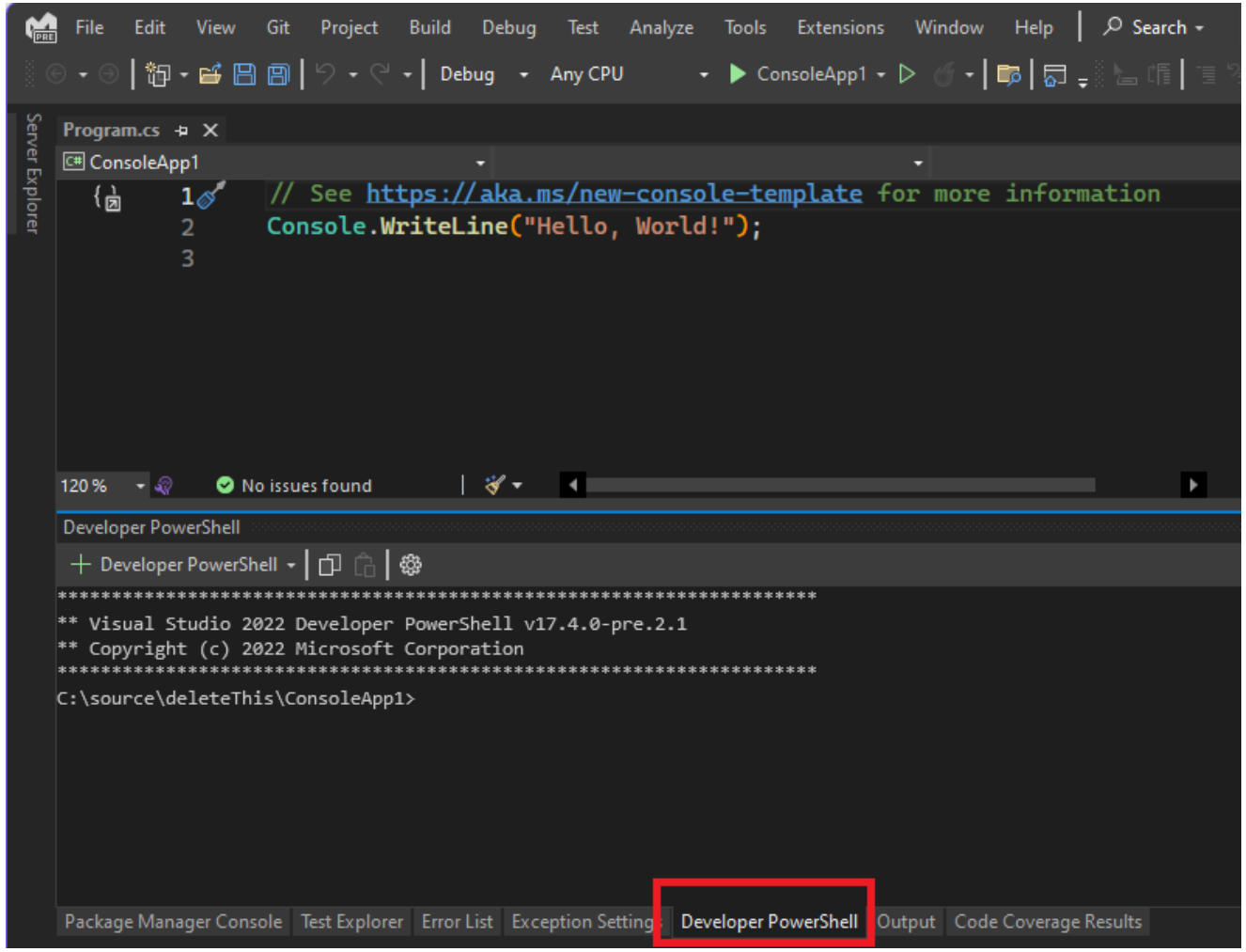


Choose a name for your app and click next.

Choose .NET 6.0 and click create

Install the OpenXML SDK and Azure.AI.TextAnalytics

Open the developer console and enter the following commands to install the Nuget packages.




```
dotnet add package DocumentFormat.OpenXml

dotnet add package Azure.AI.TextAnalytics
```

Add the usings

- Open Program.cs and delete the contents.
- Add the following usings to the top of the file.

```
using Azure;
using Azure.AI.TextAnalytics;
using DocumentFormat.OpenXml;
using DocumentFormat.OpenXml.Packaging;
using DocumentFormat.OpenXml.Wordprocessing;
```

Create an Azure TextAnalyticsClient

Below the usings, create an AzureKeyCredential and Uri with your API key and endpoint from Azure and use them to create the TextAnalyticsClient and store the file path in a variable.

```
AzureKeyCredential credentials = new AzureKeyCredential("<API Key goes here>");

Uri endpointUri = new Uri("<API endpoint goes here>");

TextAnalyticsClient textAnalyticsClient = new TextAnalyticsClient(endpointUri, credentials);

// Code on next slide goes here
```

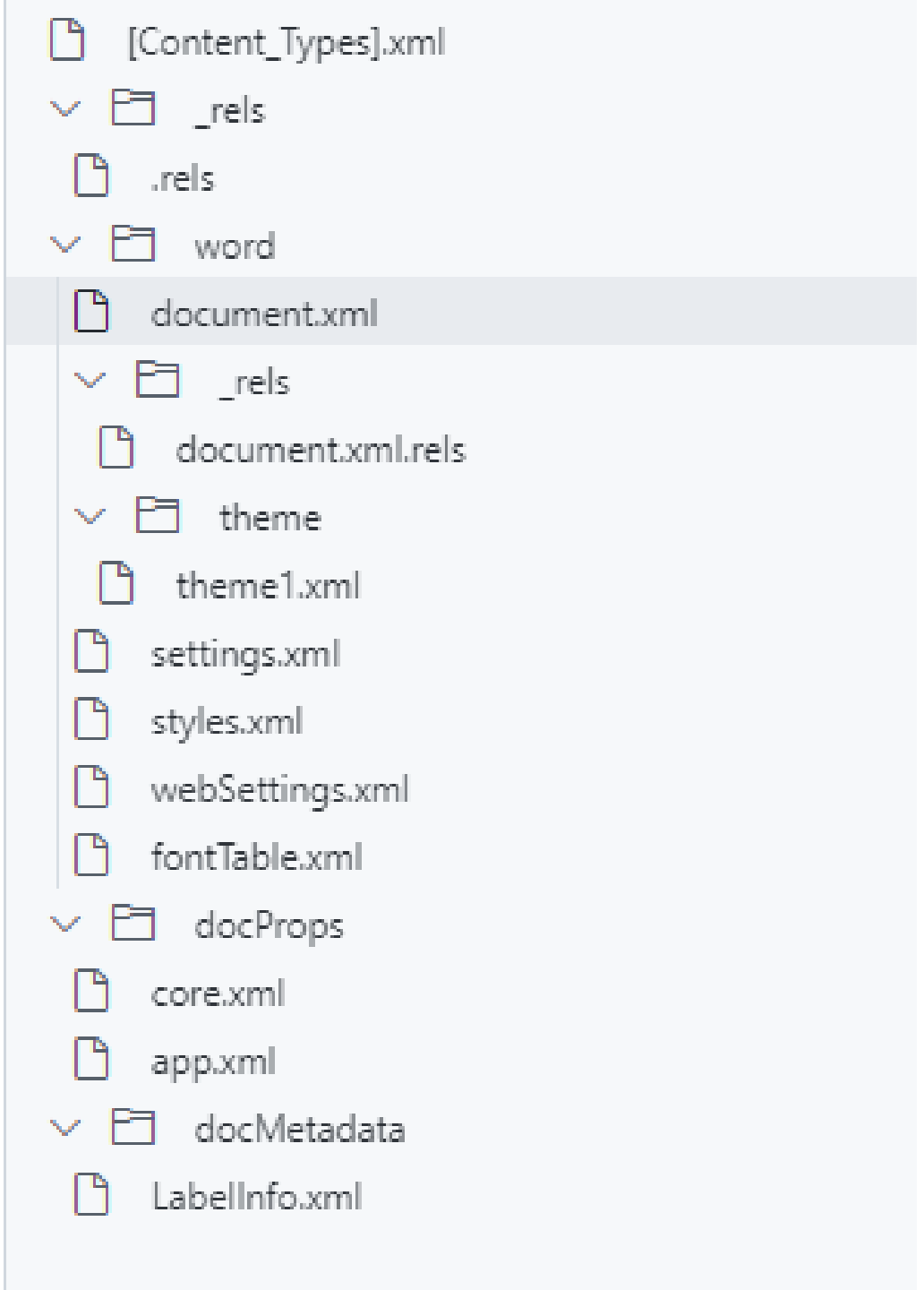
Open a Word Document and Clone It

```
string filePath = "<absolute path to .docx file with PII>";

using (WordprocessingDocument doc = WordprocessingDocument.Open(filePath, false))
{
    WordprocessingDocument? newDoc = doc.Clone() as WordprocessingDocument;
    doc.Close();
    List<Paragraph>? paragraphs =
newDoc?.MainDocumentPart?.Document?.Body?.ChildElements?.OfType<Paragraph>().ToList();

    // Code for next step goes here
}
```

How the Office Open XML File Format Works

Microsoft Office implements ISO/IEC-29500 Office Open XML File Formats

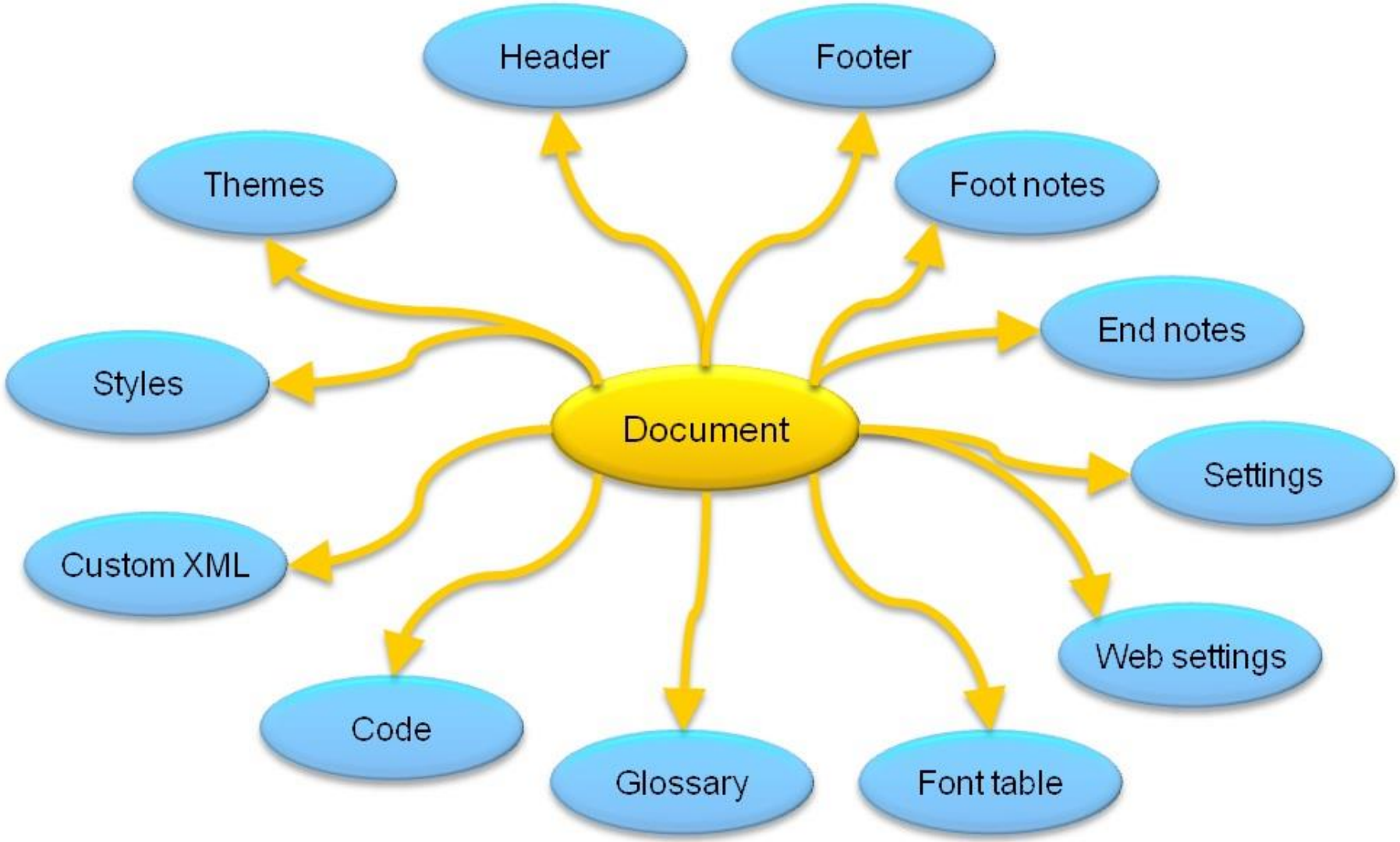
You can unzip a Word file to see the underlying xml



The Main Document Part

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Hello World</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

Typical Document Structure



The document.xml Part

```
List<Paragraph>? paragraphs =
newDoc?.MainDocumentPart?.Document?.Body?.ChildElements?.OfType<Paragraph>().ToList();
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Hello World</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

The Paragraph Element in XML

```
<w:p>
   <w:r>
      <w:t>Hello</w:t>
   </w:r>
   <w:r>
      <w:t> World</w:t>
   </w:r>
</w:p>
```

Loop over the paragraphs

- paragraphs could be null, so use the is operator to check for null
- Then loop over the paragraphs list

```
if (paragraphs is not null)
{
        foreach (Paragraph p in paragraphs)
        {
                // Code on next slide goes here
        }
}
```

Get a list of the Run elements

- Create a list of runs for each paragraph and store it in a variable called runs.
- The list could be null so use the in operator again to check for null before looping over runs

```
List<Run>? runs = p.ChildElements?.OfType<Run>().ToList();

if (runs is not null)
{
        foreach (Run r in runs)
        {
                // Next code goes here
        }
{
```

Check each run for PII and add the redacted text to the cloned document

```
if (!String.IsNullOrWhiteSpace(r.InnerText))
{
        PiiEntityCollection piiEntities =
        (await textAnalyticsClient.RecognizePiiEntitiesAsync(r.InnerText)).Value;

        if (piiEntities.Count > 0)
        {
                r.RemoveAllChildren<Text>();
                r.AppendChild(new Text(piiEntities.RedactedText)
                        { Space = SpaceProcessingModeValues.Preserve });
        }
}
```

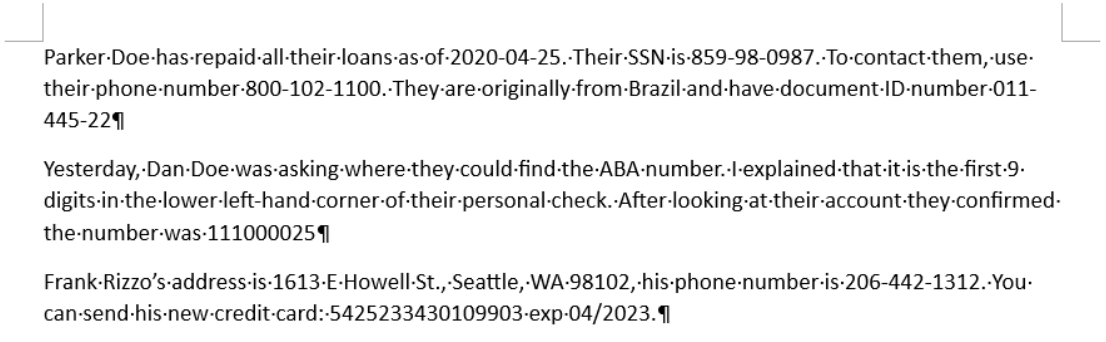
Save the new document with a new name

- At the end of the using statement use this code to create a new file name and save the file

```
if (newDoc is not null)
{
  string newFilePath = Path.Combine(
  Path.GetDirectoryName(filePath) ?? String.Empty,
  $"{Path.GetFileNameWithoutExtension(filePath)}_redacted{Path.GetExtension(filePath)}");

  newDoc.SaveAs(newFilePath);
  newDoc.Close();
}
```

The Test Document

This is the result when the app is used to process the sample test .docx file from the GitHub repo

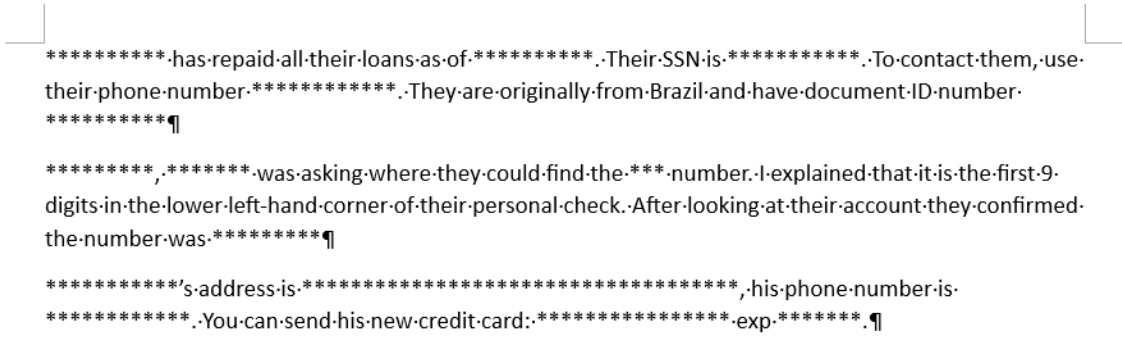


Parker·Doe·has·repaid·all·their·loans·as·of·2020-04-25.·Their·SSN·is·859-98-0987.·To·contact·them,·use·their·phone·number·800-102-1100.·They·are·originally·from·Brazil·and·have·document·ID·number·011-445-22¶

Yesterday,·Dan·Doe·was·asking·where·they·could·find·the·ABA·number.·I·explained·that·it·is·the·first·9·digits·in·the·lower·left-hand·corner·of·their·personal·check.·After·looking·at·their·account·they·confirmed·the·number·was·111000025¶

Frank·Rizzo's·address·is·1613·E·Howell·St.,·Seattle,·WA·98102,·his·phone·number·is·206-442-1312.·You·can·send·his·new·credit·card:·5425233430109903·exp·04/2023.¶

Before



**********·has·repaid·all·their·loans·as·of·**********.·Their·SSN·is·**********.·To·contact·them,·use·their·phone·number·**********.·They·are·originally·from·Brazil·and·have·document·ID·number·**********¶

*********,·*******·was·asking·where·they·could·find·the·***·number.·I·explained·that·it·is·the·first·9·digits·in·the·lower·left-hand·corner·of·their·personal·check.·After·looking·at·their·account·they·confirmed·the·number·was·*********¶

**********'s·address·is·**********,·his·phone·number·is·**********.·You·can·send·his·new·credit·card:·**********.exp.*******.¶

After

Additional Information

In addition to the redacted text, the PIIEntityCollection has additional information that can be logged to the console by adding this code.

```
foreach (PiiEntity entity in piiEntities)
{
        Console.WriteLine($"  Text: {entity.Text}");
        Console.WriteLine($"  Category: {entity.Category}");

        Console.WriteLine($"  SubCategory:
        {(!string.IsNullOrEmpty(entity.SubCategory) ? entity.SubCategory :
        String.Empty)}");

        Console.WriteLine($"  Confidence score: {entity.ConfidenceScore}");
}
```

Additional Information

The PIIEntityCollection also provides Category, SubCategory, and the Confidence score.

```
Text: Frank Rizzo
Category: Person
SubCategory:
Confidence score: 0.99
```

Some text has a high confidence score

```
Text: ABA
Category: Organization
SubCategory: Sports
Confidence score: 0.63
```

Other text the confidence is much lower

Next Steps

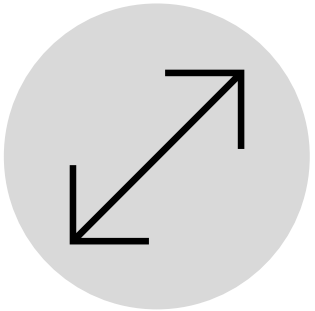
Azure Cognitive Services

- Azure Cognitive Services documetation | Microsoft Learn
- Overview of Responsible use of AI - Azure Cognitive Services | Microsoft Learn

File Format Support

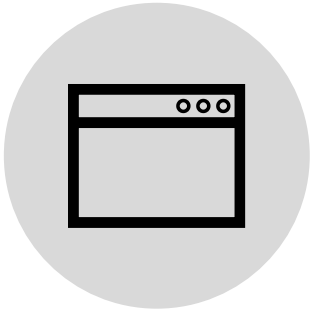
[MS-OFFPROTLP]: Office Protocols | Microsoft Learn	Read the documentation
Microsoft supported products on Q&A | Microsoft Learn	Use the tag “openspecs-office” for questions about file formats
Email DocHelp@Microsoft.com	Connect with Microsoft engineers

Open XML SDK



[Welcome to the Open XML SDK 2.5 for Office | Microsoft Learn]

[Open-XML-SDK/samples (github.com)]



[OfficeDev/Open-XML-SDK: Open XML SDK by Microsoft (github.com)]

[OOXML Viewer VSCode Extension]

End of Module

Microsoft