



EBook Gratuito

APPENDIMENTO

openxml

Free unaffiliated eBook created from
Stack Overflow contributors.

#openxml

Sommario

Di.....	1
Capitolo 1: Iniziare con openxml.....	2
Osservazioni.....	2
Examples.....	2
Installazione di OpenXML SDK e strumento di produttività sul tuo computer.....	2
Crea un nuovo foglio di calcolo con OpenXML.....	2
Utilizzo di Open XML SDK 2.5 Strumento di produttività.....	4
Capitolo 2: Come aggiungere un'immagine a un documento di Word.....	7
introduzione.....	7
Osservazioni.....	7
Examples.....	7
Aggiungi l'immagine alla struttura di OpenXml.....	7
Fare riferimento all'immagine nel documento di Word.....	7
Capitolo 3: Crea un nuovo documento di Word con Open XML.....	10
introduzione.....	10
Examples.....	10
Ciao mondo.....	10
Capitolo 4: Inserisci l'immagine in una "forma in linea" incorporata nei documenti word.....	13
introduzione.....	13
Examples.....	13
Aggiungi i seguenti spazi dei nomi OpenXML alla tua classe.....	13
Aprire il documento e aggiungere l'oggetto imagePart per fare riferimento all'immagine che.....	13
Ottieni un riferimento di un oggetto Blip.....	13
Aggiunta del riferimento dell'immagine alle forme nel documento modello.....	14
Con un riferimento alla collezione di tutte le forme, passa attraverso la collezione.....	14
Titoli di coda.....	16

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [openxml](#)

It is an unofficial and free openxml ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official openxml.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con openxml

Osservazioni

Questa sezione fornisce una panoramica su cosa sia openxml e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di openxml e collegarsi agli argomenti correlati. Poiché la documentazione di openxml è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Installazione di OpenXML SDK e strumento di produttività sul tuo computer

Vai al [collegamento Microsoft per il download SDK OpenXML](#) . Fai clic sul pulsante di download rosso. Nella schermata successiva fare clic sulla casella accanto a OpenXMLSDKToolV25.msi e fare clic su Avanti per avviare il download.

Una volta completato il download, avvia OpenXMLSDKToolV25.msi e segui le istruzioni sullo schermo.

Il programma di installazione posiziona i file nella seguente directory predefinita:

```
"C:\Program Files (x86)\Open XML SDK\V2.5"
```

In questa directory è presente un readme che spiega come utilizzare l'SDK e un readme per lo strumento di produttività.

Crea un nuovo foglio di calcolo con OpenXML

Questo metodo creerà un nuovo foglio di calcolo di Excel. Passare nel `fileName` che è un nome di percorso completo del file.

```
using DocumentFormat.OpenXml;
using DocumentFormat.OpenXml.Packaging;
using DocumentFormat.OpenXml.Spreadsheet;
using System;
....
void Create(string fileName)
{
    using (SpreadsheetDocument document = SpreadsheetDocument.Create(fileName,
SpreadsheetDocumentType.Workbook))
    {
        var relationshipId = "rId1";

        //build Workbook Part
        var workbookPart = document.AddWorkbookPart();
```

```

var workbook = new Workbook();
var sheets = new Sheets();
var sheet1 = new Sheet() { Name = "First Sheet", SheetId = 1, Id = relationshipId
};

sheets.Append(sheet1);
workbook.Append(sheets);
workbookPart.Workbook = workbook;

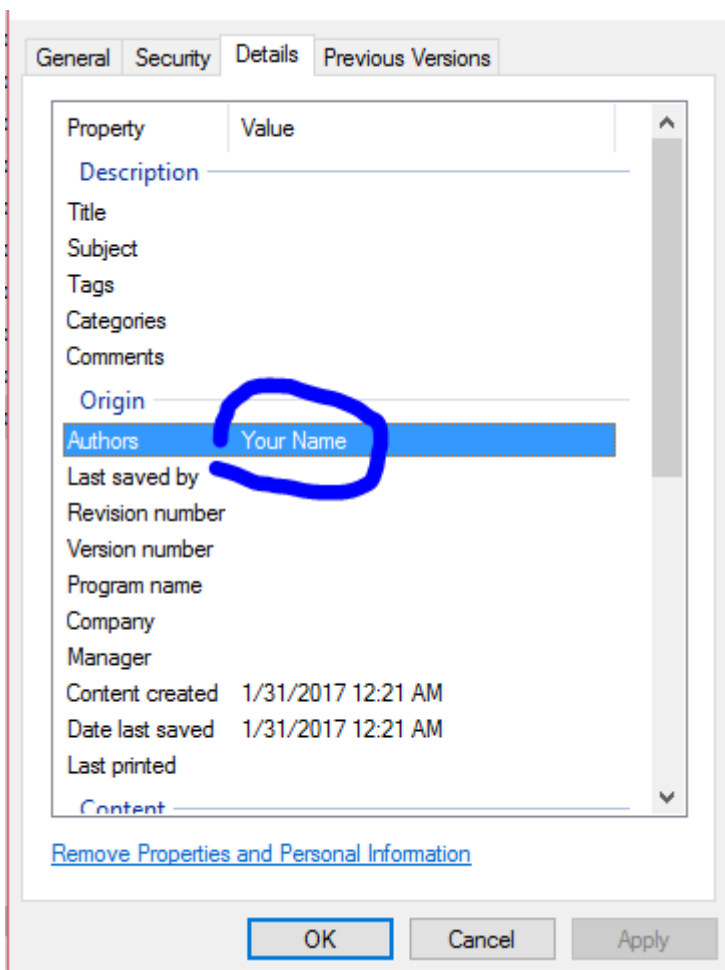
//build Worksheet Part
var workSheetPart = workbookPart.AddNewPart<WorksheetPart>(relationshipId);
var workSheet = new Worksheet();
workSheet.Append(new SheetData());
workSheetPart.Worksheet = workSheet;

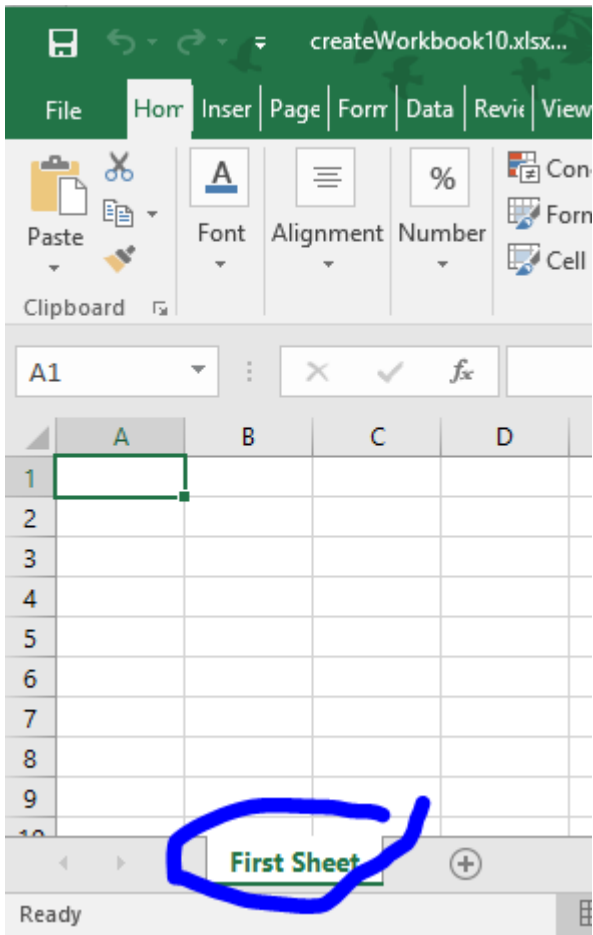
//add document properties
document.PackageProperties.Creator = "Your Name";
document.PackageProperties.Created = DateTime.UtcNow;
}

```

Per questo progetto, assicurati di includere il riferimento a `DocumentFormat.OpenXml`. Questo si trova nel percorso specificato nell'Installazione di OpenXML Example.

Il foglio di calcolo verrà creato con il **tuo nome** come autore e il primo foglio di lavoro denominato **Primo foglio**.





Utilizzo di Open XML SDK 2.5 Strumento di produttività

Leggere le specifiche per i formati di documento in OpenXML può essere un processo che richiede tempo. A volte vuoi solo vedere come produrre una determinata funzione in un documento di testo. Lo strumento di produttività Open XML SDK 2.5 per Microsoft Office (OpenXmlSdkTool.exe) fa proprio questo. Le sue caratteristiche principali sono:

- Vedi la struttura di un file - quali parti di xml contiene
- Naviga su xml in ognuna di queste parti
- Genera codice c # per produrre la parte selezionata del documento
- Link alle specifiche del formato file che descrivono maggiori dettagli
- Documentare la convalida di OpenXML

Per un semplice 'Hello world.docx' sembra questo:

Document Explorer

- {} Hello world.docx
 - [] /docProps/app.xml
 - > <> ap:Properties (Properties)
 - [] /docProps/core.xml
 - ▲ [] /word/document.xml
 - > [] /word/webSettings.xml
 - > [] /word/settings.xml
 - > [] /word/styles.xml
 - > [] /word/theme/theme1.xml
 - > [] /word/fontTable.xml
 - ▲ <> w:document (Document)
 - ▲ <> w:body (Body)
 - ▲ <> w:p (Paragraph)
 - ▲ <> w:r (Run)
 - <> w:t (Text)
 - <> w:bookmarkStart (BookmarkStart)
 - <> w:bookmarkEnd (BookmarkEnd)
 - > <> w:sectPr (SectionProperties)

Reflected Code

```
<w:p w:rsidR
  <w:r>
    <w:t>Hel
  </w:r>
  <w:bookmark
  <w:bookmark
</w:p>
```

```
using Document
using Document
```

```
namespace Ge
{
  public c
  {
    // C
    publ
    {
```

Il riquadro a sinistra mostra la struttura del documento. Il riquadro in alto a destra mostra l'xml corrispondente alla selezione nell'albero, e infine il riquadro in basso a destra mostra un codice generato per produrre l'xml visualizzato sopra di esso.

Ciò consente di esplorare molto facilmente una determinata funzionalità:

- Produrre un documento di esempio (fx un documento word)
- Apri il documento in Strumento produttività
- Usa "Rifletti codice" per generare codice

L'SDK può essere scaricato da <https://www.microsoft.com/en-us/download/details.aspx?id=30425> - scarica e installa entrambi i pacchetti msi. Dopo l'installazione, utilizzare OpenXMLSdkTool.exe installato in "C: \ Programmi (x86) \ Apri SDK XML \ V2.5 \ tool".

Leggi Iniziare con openxml online: <https://riptutorial.com/it/openxml/topic/6967/iniziare-con-openxml>

Capitolo 2: Come aggiungere un'immagine a un documento di Word.

introduzione

L'inserimento di un'immagine in un documento word utilizzando OpenXml richiede due azioni: aggiungi l'immagine all'interno di openxml e fai riferimento all'immagine nel documento

Osservazioni

se si aggiunge l'immagine alla struttura openxml senza farla riferimento nel documento di Word, la prossima volta che si "apre / salva" il documento, il file dell'immagine verrà eliminato.

Word elimina tutti i riferimenti orfani. Quindi assicurati di aggiungere l'immagine nel documento di Word, altrimenti dovrai ripetere tutti i passaggi.

Examples

Aggiungi l'immagine alla struttura di OpenXml

```
private string AddGraph(WordprocessingDocument wpd, string filepath)
{
    ImagePart ip = wpd.MainDocumentPart.AddImagePart(ImagePartType.Jpeg);
    using (FileStream fs = new FileStream(filepath, FileMode.Open))
    {
        if (fs.Length == 0) return string.Empty;
        ip.FeedData(fs);
    }

    return wpd.MainDocumentPart.GetIdOfPart(ip);
}
```

In questo caso usiamo un FileStream per recuperare l'immagine, ma feedData (Stream) è in attesa di qualsiasi tipo di Stream.

Fare riferimento all'immagine nel documento di Word

```
private void InsertImage(WordprocessingDocument wpd, OpenXmlElement parent, string filepath)
{
    string relationId = AddGraph(wpd, filepath);
    if (!string.IsNullOrEmpty(relationId))
    {
        Size size = new Size(800, 600);

        Int64Value width = size.Width * 9525;
        Int64Value height = size.Height * 9525;
    }
}
```

```

var draw = new Drawing(
    new DW.Inline(
        new DW.Extent() { Cx = width, Cy = height },
        new DW.EffectExtent()
        {
            LeftEdge = 0L,
            TopEdge = 0L,
            RightEdge = 0L,
            BottomEdge = 0L
        },
        new DW.DocProperties()
        {
            Id = (UInt32Value)1U,
            Name = "my image name"
        },
        new DW.NonVisualGraphicFrameDrawingProperties(new A.GraphicFrameLocks() {
NoChangeAspect = true })),
    new A.Graphic(
        new A.GraphicData(
            new PIC.Picture(
                new PIC.NonVisualPictureProperties(
                    new PIC.NonVisualDrawingProperties()
                    {
                        Id = (UInt32Value)0U,
                        Name = relationId
                    },
                    new PIC.NonVisualPictureDrawingProperties()),
                new PIC.BlipFill(
                    new A.Blip(
                        new A.BlipExtensionList(
                            new A.BlipExtension() { Uri = "{28A0092B-C50C-
407E-A947-70E740481C1C}" })
                        )
                    {
                        Embed = relationId,
                        CompressionState =
                            A.BlipCompressionValues.Print
                    },
                    new A.Stretch(
                        new A.FillRectangle()),
                    new PIC.ShapeProperties(
                        new A.Transform2D(
                            new A.Offset() { X = 0L, Y = 0L },
                            new A.Extents() { Cx = width, Cy = height
}),
                        new A.PresetGeometry(new
A.AdjustValueList() { Preset = A.ShapeTypeValues.Rectangle }))) { Uri =
"http://schemas.openxmlformats.org/drawingml/2006/picture" })
                )
            {
                DistanceFromTop = (UInt32Value)0U,
                DistanceFromBottom = (UInt32Value)0U,
                DistanceFromLeft = (UInt32Value)0U,
                DistanceFromRight = (UInt32Value)0U,
                EditId = "50D07946"
            }
        });
    parent.Append(draw);
}
}

```

In questo esempio ho impostato una dimensione statica di 800 * 600 ma è possibile impostare qualsiasi dimensione necessaria

Leggi [Come aggiungere un'immagine a un documento di Word](https://riptutorial.com/it/openxml/topic/9397/come-aggiungere-un-immagine-a-un-documento-di-word). online:

[https://riptutorial.com/it/openxml/topic/9397/come-aggiungere-un-immagine-a-un-documento-di-word-](https://riptutorial.com/it/openxml/topic/9397/come-aggiungere-un-immagine-a-un-documento-di-word)

Capitolo 3: Crea un nuovo documento di Word con Open XML

introduzione

Lo standard di markup del documento OpenXML è un formato basato su XML che consente soluzioni su molte piattaforme software e sistemi operativi.

Examples

Ciao mondo

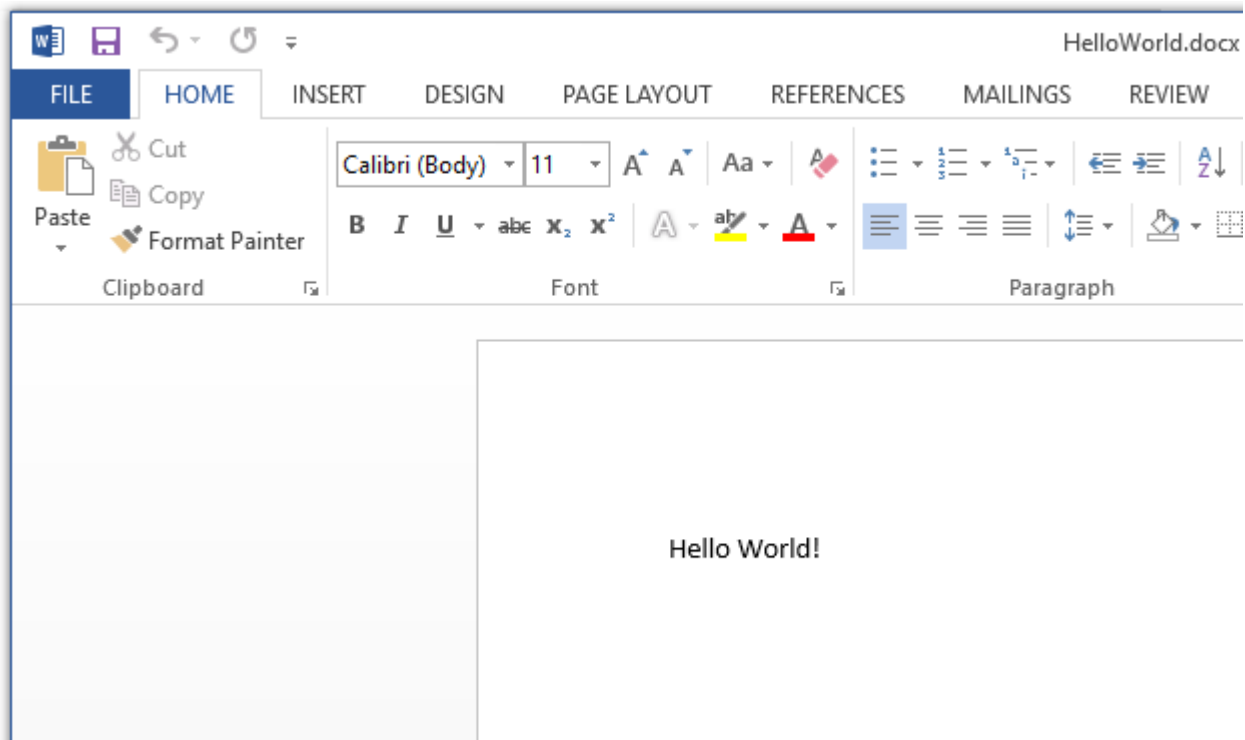
Innanzitutto, crea un nuovo progetto di console utilizzando Visual Studio e aggiungi i seguenti file .dll al tuo progetto:

```
DocumentFormat.OpenXml  
WindowsBase
```

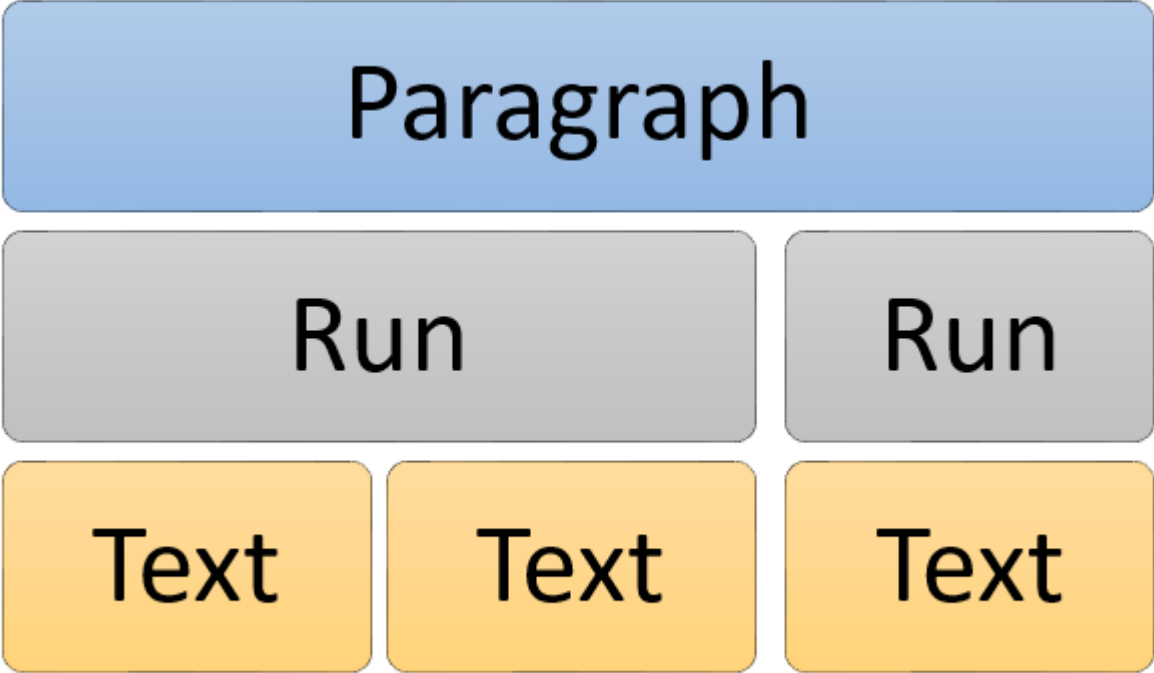
Successivamente, compila ed esegui il seguente codice:

```
static void Main(string[] args)  
{  
    // Create a Wordprocessing document.  
    using ( WordprocessingDocument package = WordprocessingDocument.Create("HelloWorld.docx",  
WordprocessingDocumentType.Document))  
    {  
        // Add a new main document part.  
        package.AddMainDocumentPart();  
  
        // Create the Document DOM.  
        package.MainDocumentPart.Document =  
            new Document(  
                new Body(  
                    new Paragraph(  
                        new Run(  
                            new Text("Hello World!")))))));  
  
        // Save changes to the main document part.  
        package.MainDocumentPart.Document.Save();  
    }  
}
```

Sotto la tua cartella `\bin\Debug` dovresti avere il tuo primo documento WordprocessingML:



Il testo che abbiamo aggiunto nell'esempio sopra è memorizzato sotto la parte del documento principale. All'interno della parte del documento principale c'è l'elemento del **documento** che consente a un **corpo di** elemento figlio di memorizzare il testo che rende il nostro documento. Esistono due gruppi principali di contenuti per il corpo del documento, il livello di blocco (*paragrafi e tabelle*) e il contenuto in linea (*corse e testo*). Il contenuto a livello di blocco fornisce la struttura principale e contiene contenuto in linea. Per comprendere l'esempio sopra, dobbiamo prima capire la gerarchia testuale in WordprocessingML. Un paragrafo è diviso in diverse esecuzioni. Una corsa è l'elemento di livello più basso a cui può essere applicata la formattazione. La corsa viene nuovamente suddivisa in vari elementi di testo.



Leggi [Crea un nuovo documento di Word con Open XML online:](https://riptutorial.com/it/openxml/topic/9833/crea-un-nuovo-documento-di-word-con-open-xml)
<https://riptutorial.com/it/openxml/topic/9833/crea-un-nuovo-documento-di-word-con-open-xml>

Capitolo 4: Inserisci l'immagine in una "forma in linea" incorporata nei documenti word

introduzione

Inserisci un'immagine in forme di documenti di MS Word come Rettangoli e ovali.

Questa documentazione presume che tu sappia come inserire un'immagine in un documento word, aprire e chiudere un documento word usando OpenXML

Examples

Aggiungi i seguenti spazi dei nomi OpenXML alla tua classe

```
using System;
using System.Collections.Generic;
using System.Linq;
using DocumentFormat.OpenXml;
using A = DocumentFormat.OpenXml.Drawing;
using DW = DocumentFormat.OpenXml.Drawing.Wordprocessing;
using PIC = DocumentFormat.OpenXml.Drawing.Pictures;
using DocumentFormat.OpenXml.Drawing.Wordprocessing;
using Wps = DocumentFormat.OpenXml.Office2010.Word.DrawingShape;
```

Aprire il documento e aggiungere l'oggetto imagePart per fare riferimento all'immagine che si desidera inserire nella forma

Ora apri il documento usando OpenXML, devi aggiungere un imagePart che fa riferimento all'oggetto immagine all'oggetto MainDocumentPart utilizzando un flusso di file e ottieni l'ID dell'immagine

```
string temp;
MainDocumentPart mainPart = document.MainDocumentPart;
ImagePart imagePart = mainPart.AddImagePart(ImagePartType.Bmp);

using (FileStream stream = new FileStream(barcodepath,
    FileMode.Open))
{
    imagePart.FeedData(stream);
}

temp = mainPart.GetIdOfPart(imagePart);
```

Ottieni un riferimento di un oggetto Blip

In Office OpenXML, un'immagine che viene inserita in un documento word è considerata un oggetto o elemento "Blip". La classe è derivata da [DocumentFormat.OpenXml.Drawing](#) il Blip deve

avere un valore di incorporamento che è un ID di imagePart. L'oggetto Blip viene quindi [inserito](#) all'interno di un oggetto / elemento [BlipFill](#) e anche all'interno di un oggetto / elemento [graphicData](#) che a sua volta entra in un elemento oggetto [grafico](#) . Sono abbastanza sicuro che ora hai realizzato che tutto funziona come un albero XML. Esempio di albero XML aperto di seguito.

```
<a:graphic xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main">
  <a:graphicData
uri="http://schemas.microsoft.com/office/word/2010/wordprocessingShape">
  <wps:wsp>
    <wps:cNvSpPr>
      <a:spLocks noChangeArrowheads="1" />
    </wps:cNvSpPr>
    <wps:spPr bwMode="auto">
      <a:xfrm>
        <a:off x="0" y="0" />
        <a:ext cx="1234440" cy="1234440" />
      </a:xfrm>
      <a:prstGeom prst="roundRect">
        <a:avLst>
          <a:gd name="adj" fmla="val 16667" />
        </a:avLst>
      </a:prstGeom>
      <a:blipFill dpi="0" rotWithShape="1">
        <a:blip r:embed="Raade88ffea8d4c1b" />
        <a:stretch>
          <a:fillRect l="10000" t="10000" r="10000" b="10000" />
        </a:stretch>
      </a:blipFill>
    </wps:spPr>
    <wps:bodyPr rot="0" vert="horz" wrap="square" lIns="91440"
tIns="45720" rIns="91440" bIns="45720" anchor="t" anchorCtr="0" upright="1">
      <a:noAutofit />
    </wps:bodyPr>
  </wps:wsp>
</a:graphicData>
</a:graphic>
```

Aggiunta del riferimento dell'immagine alle forme nel documento modello.

Ora hai un riferimento dell'immagine. Inserisci l'immagine nelle forme nel documento modello. Per fare ciò, dovrai usare qualche LINQ per scorrere il documento e ottenere un riferimento a tutte le forme nel documento. L'elemento wp: spPr che vedi nel codice XML di cui sopra è l'elemento xml per le forme in un documento. La classe C # equivalente è WordprocessingShape

```
IEnumerable<DocumentFormat.OpenXml.Office2010.Word.DrawingShape.WordprocessingShape> shapes2 =
document.MainDocumentPart.document.Body.Descendants<DocumentFormat.OpenXml.Office2010.Word.DrawingShape>
```

Con un riferimento alla collezione di tutte le forme, passa attraverso la collezione.

Ora che hai una collezione di tutti i riferimenti di forma nel documento. Passa attraverso la collezione con un foreach, e attraverso ogni iterazione crea un oggetto Blip. Imposta il valore di incorporamento dell'oggetto Blip sul riferimento dell'ID dell'immagine acquisito in precedenza dalla

parte dell'immagine. Crea anche un oggetto Stretch e un oggetto FillRectangle (questi non sono realmente necessari, sono solo usati per il corretto allineamento dell'immagine). E aggiungi ciascuno al suo oggetto padre come l'equivalente dell'albero XML.

```
foreach (DocumentFormat.OpenXml.Office2010.Word.DrawingShape.WordprocessingShape sp in
shapes2)
    {
        //Wps.ShapeProperties shapeProperties1 =
        A.BlipFill blipFill1 = new A.BlipFill() { Dpi =
(UInt32Value)0U, RotateWithShape = true };
        A.Blip blip1 = new A.Blip() { Embed = temp };

        A.Stretch stretch1 = new A.Stretch();
        A.FillRectangle fillRectangle1 = new A.FillRectangle() {
Left = 10000, Top = 10000, Right = 10000, Bottom = 10000 };
        Wps.WordprocessingShape wordprocessingShape1 = new
Wps.WordprocessingShape();

        stretch1.Append(fillRectangle1);
        blipFill1.Append(blip1);
        blipFill1.Append(stretch1);
        Wps.ShapeProperties shapeProperties1 =
sp.Descendants<Wps.ShapeProperties>().First();
        shapeProperties1.Append(blipFill1);
    }
```

Leggi [Inserisci l'immagine in una "forma in linea" incorporata nei documenti word online: https://riptutorial.com/it/openxml/topic/9660/inserisci-l-immagine-in-una--forma-in-linea--incorporata-nei-documenti-word](https://riptutorial.com/it/openxml/topic/9660/inserisci-l-immagine-in-una--forma-in-linea--incorporata-nei-documenti-word)

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con openxml	Community , Daniel Brixen , Mandar Badve , Taterhead
2	Come aggiungere un'immagine a un documento di Word.	Maxime Porté
3	Crea un nuovo documento di Word con Open XML	FortyTwo
4	Inserisci l'immagine in una "forma in linea" incorporata nei documenti word	Dankwansere