**Как: Вставка изображения в документ word (Open XML SDK)**

**Office 2013**

Настоящая документация является предварительной и может быть изменена.

В этом разделе показано, как использовать классы в пакете Пакет Open XML SDK 2.5 для Office для добавления изображения в документ Word программными средствами.

Для компиляции кода, представленного в этом разделе, требуются следующие директивы сборки.

C#

[VB](http://msdn.microsoft.com/ru-ru/library/office/bb497430%28v=office.15%29.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-1)

using System.IO;

using DocumentFormat.OpenXml;

using DocumentFormat.OpenXml.Packaging;

using DocumentFormat.OpenXml.Wordprocessing;

using A = DocumentFormat.OpenXml.Drawing;

using DW = DocumentFormat.OpenXml.Drawing.Wordprocessing;

using PIC = DocumentFormat.OpenXml.Drawing.Pictures;

[**Открытие существующего документа для изменения**](javascript:void(0))

Чтобы открыть существующий документ, необходимо создать экземпляр класса [WordprocessingDocument](http://msdn.microsoft.com/ru-ru/library/office/documentformat.openxml.packaging.wordprocessingdocument%28v=office.15%29.aspx), как показано в следующей инструкции using. В той же инструкции откройте файл текстового документа, указав путь к файлу в методе [Open(String, Boolean)](http://msdn.microsoft.com/ru-ru/library/office/cc562234%28v=office.15%29.aspx) и установив для логического параметра значение true, чтобы разрешить изменение документа.

C#

[VB](http://msdn.microsoft.com/ru-ru/library/office/bb497430%28v=office.15%29.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

using (WordprocessingDocument wordprocessingDocument =

WordprocessingDocument.Open(filepath, true))

{

// Insert other code here.

}

Инструкцию using рекомендуется использовать вместо традиционной последовательности .Open, .Save, .Close. Это позволяет обеспечить автоматический вызов метода Dispose (внутреннего метода, используемого пакетом SDK Open XML для очистки ресурсов) при достижении закрывающей скобки. Блок, следующий за инструкцией using, создает область для объекта, создаваемого или именуемого в инструкции using, в данном случае — *wordprocessingDocument*. Так как класс WordprocessingDocument в пакете SDK Open XML автоматически сохраняет и закрывает объект как часть реализации System.IDisposable, а также поскольку метод Dispose вызывается автоматически при выходе из блока, не нужно явно вызывать методы Save и Close, если используется инструкция using.

[**XML-представление графического объекта**](javascript:void(0))

В спецификации [ISO/IEC 29500 (Возможно, на английском языке)](http://go.microsoft.com/fwlink/?linkid=194337&clcid=0x419) содержатся следующие сведения об элементе данных графического объекта.

Этот элемент определяет ссылку на графический объект в документе. Графический объект добавляется авторами документа, которые сохраняют в документе эти данные.[*Примечание*. В зависимости от используемого типа графического объекта некоторые приложения для создания графики с поддержкой платформы OOXML могут не поддерживать отображение этого графического объекта. *конец примечания*]© ISO/IEC29500: 2008.

Содержимое этого элемента определено в следующем фрагменте схемы XML

XML

<complexType name="CT\_GraphicalObjectData">

<sequence>

<any minOccurs="0" maxOccurs="unbounded" processContents="strict"/>

</sequence>

<attribute name="uri" type="xsd:token"/>

</complexType>

[**Механизм работы примера кода**](javascript:void(0))

После открытия документа добавьте объект [ImagePart](http://msdn.microsoft.com/ru-ru/library/office/documentformat.openxml.packaging.imagepart%28v=office.15%29.aspx) в объект [MainDocumentPart](http://msdn.microsoft.com/ru-ru/library/office/documentformat.openxml.packaging.maindocumentpart%28v=office.15%29.aspx), используя поток файлов, как показано в следующем фрагменте кода.

C#

[VB](http://msdn.microsoft.com/ru-ru/library/office/bb497430%28v=office.15%29.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4)

MainDocumentPart mainPart = wordprocessingDocument.MainDocumentPart;

ImagePart imagePart = mainPart.AddImagePart(ImagePartType.Jpeg);

using (FileStream stream = new FileStream(fileName, FileMode.Open))

{

imagePart.FeedData(stream);

}

AddImageToBody(wordprocessingDocument, mainPart.GetIdOfPart(imagePart));

Для добавления изображения в основную часть документа сначала определит ссылку на изображение. Затем добавьте ссылку в основную часть документа. Элемент должен содержаться в [Run](http://msdn.microsoft.com/ru-ru/library/office/documentformat.openxml.wordprocessing.run%28v=office.15%29.aspx).

C#

[VB](http://msdn.microsoft.com/ru-ru/library/office/bb497430%28v=office.15%29.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-5)

// Define the reference of the image.

var element =

new Drawing(

new DW.Inline(

new DW.Extent() { Cx = 990000L, Cy = 792000L },

new DW.EffectExtent()

{

LeftEdge = 0L,

TopEdge = 0L,

RightEdge = 0L,

BottomEdge = 0L

},

new DW.DocProperties()

{

Id = (UInt32Value)1U,

Name = "Picture 1"

},

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 = "New Bitmap Image.jpg"

},

new PIC.NonVisualPictureDrawingProperties()),

new PIC.BlipFill(

new A.Blip(

new A.BlipExtensionList(

new A.BlipExtension()

{

Uri =

"{28A0092B-C50C-407E-A947-70E740481C1C}"

})

)

{

Embed = relationshipId,

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 = 990000L, Cy = 792000L }),

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"

});

// Append the reference to the body. The element should be in

// a DocumentFormat.OpenXml.Wordprocessing.Run.

wordDoc.MainDocumentPart.Document.Body.AppendChild(new Paragraph(new Run(element)));

[**Пример кода**](javascript:void(0))

В следующем коде в существующий документ Word добавляется изображение. Метод InsertAPicture можно вызвать, передав в него путь к документу Word и путь к файлу с изображением. Например, посредством следующего вызова можно вставить изображение "MyPic.jpg" в файл "Word9.docx" с указанием соответствующих путей.

C#

[VB](http://msdn.microsoft.com/ru-ru/library/office/bb497430%28v=office.15%29.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-6)

string document = @"C:\Users\Public\Documents\Word9.docx";

string fileName = @"C:\Users\Public\Documents\MyPic.jpg";

InsertAPicture(document, fileName);

После запуска кода проверьте вставленное изображение в файле "Word9.docx".

Ниже приведен полный пример кода на языках C# и Visual Basic.

C#

[VB](http://msdn.microsoft.com/ru-ru/library/office/bb497430%28v=office.15%29.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-7)

public static void InsertAPicture(string document, string fileName)

{

using (WordprocessingDocument wordprocessingDocument =

WordprocessingDocument.Open(document, true))

{

MainDocumentPart mainPart = wordprocessingDocument.MainDocumentPart;

ImagePart imagePart = mainPart.AddImagePart(ImagePartType.Jpeg);

using (FileStream stream = new FileStream(fileName, FileMode.Open))

{

imagePart.FeedData(stream);

}

AddImageToBody(wordprocessingDocument, mainPart.GetIdOfPart(imagePart));

}

}

private static void AddImageToBody(WordprocessingDocument wordDoc, string relationshipId)

{

// Define the reference of the image.

var element =

new Drawing(

new DW.Inline(

new DW.Extent() { Cx = 990000L, Cy = 792000L },

new DW.EffectExtent() { LeftEdge = 0L, TopEdge = 0L,

RightEdge = 0L, BottomEdge = 0L },

new DW.DocProperties() { Id = (UInt32Value)1U,

Name = "Picture 1" },

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 = "New Bitmap Image.jpg" },

new PIC.NonVisualPictureDrawingProperties()),

new PIC.BlipFill(

new A.Blip(

new A.BlipExtensionList(

new A.BlipExtension()

{ Uri =

"{28A0092B-C50C-407E-A947-70E740481C1C}" })

)

{ Embed = relationshipId,

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 = 990000L, Cy = 792000L }),

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" });

// Append the reference to body, the element should be in a Run.

wordDoc.MainDocumentPart.Document.Body.AppendChild(new Paragraph(new Run(element)));

}