

# Namespace FileFormat.Slides

## Classes

### [Image](#)

This class represents the image within a slide.

### [Presentation](#)

Represents the presentation document.

### [Slide](#)

Represents the slide object within a presentation

### [StyledList](#)

### [TextSegment](#)

This class represents the text segment within a paragraph.

### [TextShape](#)

This class represents the text shape within a slide.

# Class Image

Namespace: [FileFormat.Slides](#)

Assembly: FileFormat.Slides.dll








This class represents the image within a slide.

```
public class Image
```

## Inheritance

[object](#)  ← Image

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

# Constructors

## Image()

Blank constructor to initialize the image object

```
public Image()
```

## Image(string)

Initialize the image object

```
public Image(string imagePath)
```

## Parameters

**imagePath** [string](#) 

Image path as string

# Properties

## Facade

Property to get or set the ImageFacade instance.

```
public ImageFacade Facade { get; set; }
```

Property Value

[ImageFacade](#)

## Height

Property to get or set the height of an image.

```
public double Height { get; set; }
```

Property Value

[double](#)<sup>↗</sup>

## ImageIndex

Property to get or set the image index within the slide.

```
public int ImageIndex { get; set; }
```

Property Value

[int](#)<sup>↗</sup>

## ImagePath

Property to get or set the image path.

```
public string ImagePath { get; set; }
```

Property Value

[string](#)↗

## Name

Property to get or set the image index within the slide.

```
public string Name { get; set; }
```

Property Value

[string](#)↗

## Width

Property to get or set the width of an image.

```
public double Width { get; set; }
```

Property Value

[double](#)↗

## X

Property to get or set the X coordinate of an image.

```
public double X { get; set; }
```

Property Value

[double](#)↗

## Y

Property to get or set the Y coordinate of an image.

```
public double Y { get; set; }
```

Property Value

[double](#)<sup>↗</sup>

## Methods

### GetImages(List<ImageFacade>)

Method to get the list of the images within a slide

```
public static List<Image> GetImages(List<ImageFacade> imageFacades)
```

Parameters

**imageFacades** [List](#)<sup>↗</sup> [<ImageFacade>](#)

An object of ImageFacade.

Returns

[List](#)<sup>↗</sup> [<Image>](#)

### Remove()

Method to remove the image.

```
public void Remove()
```

### Update()

```
public void Update()
```

# Class Presentation

Namespace: [FileFormat.Slides](#)

Assembly: FileFormat.Slides.dll








Represents the presentation document.

```
public class Presentation
```

## Inheritance

[object](#)  ← Presentation

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

## Methods

### AppendSlide(Slide)

This method is responsible to append a slide.

```
public void AppendSlide(Slide slide)
```

## Parameters

**slide** [Slide](#)

An object of a slide

### Create(string)

Static method to instantiate a new object of Presentation class.

```
public static Presentation Create(string FilePath)
```

## Parameters

**FilePath** [string](#)

Presentation path as string

## Returns

[Presentation](#)

An instance of Presentation object

## Examples

```
Presentation presentation = Presentation.Create("D:\\AsposeSampleResults\\test2.pptx");
TextShape shape = new TextShape();
shape.Text = "Title: Here is my first title From FF";
TextShape shape2 = new TextShape();
shape2.Text = "Body : Here is my first title From FF";
// First slide
Slide slide = new Slide();
slide.AddTextShapes(shape);
slide.AddTextShapes(shape2);
// 2nd slide
Slide slide1 = new Slide();
slide1.AddTextShapes(shape);
slide1.AddTextShapes(shape2);
// Adding slides
presentation.AppendSlide(slide);
presentation.AppendSlide(slide1);
presentation.Save();
```

## ExtractAndSaveImages(string)

Extract and save images of a presentation into a director

```
public void ExtractAndSaveImages(string outputFolder)
```

## Parameters

**outputFolder** [string](#)



Folder path as string

## GetSlides()

Method to get the list of all slides of a presentation

```
public List<Slide> GetSlides()
```

Returns

[List](#) [<Slide>](#)

Examples

```
Presentation presentation = Presentation.Open("D:\\AsposeSampleData\\sample.pptx");  
var slides = presentation.GetSlides();  
var slide = slides[0];  
...
```

## InsertSlideAt(int, Slide)

Method to insert a slide at a specific index

```
public void InsertSlideAt(int index, Slide slide)
```

Parameters

**index** [int](#)

Index of a slide

**slide** [Slide](#)

A slide object

## Open(string)

Static method to load an existing presentation.

```
public static Presentation Open(string FilePath)
```

## Parameters

FilePath [string](#)

Presentation path as string

## Returns

[Presentation](#)

## Examples

```
Presentation presentation = Presentation.Open("D:\\AsposeSampleData\\sample.pptx");
TextShape shape1 = new TextShape();
shape1.Text = "Title: Here is my first title From FF";
TextShape shape2 = new TextShape();
shape2.Text = "Body : Here is my first title From FF";
// New slide
Slide slide = new Slide();
slide.AddTextShapes(shape1);
slide.AddTextShapes(shape2);
// Adding slide
presentation.AppendSlide(slide);
presentation.Save();
```

## RemoveSlide(int)

Method to remove a slide at a specific index

```
public string RemoveSlide(int slideIndex)
```

## Parameters

slideIndex [int](#)

Index of a slide

## Returns

[string](#) 

## Examples

```
Presentation presentation = Presentation.Open("D:\\AsposeSampleData\\sample.pptx");  
var confirmation = presentation.RemoveSlide(0);  
Console.WriteLine(confirmation);  
presentation.Save();
```

## Save()

Method to save the new or changed presentation.

```
public void Save()
```

# Class Slide

Namespace: [FileFormat.Slides](#)

Assembly: FileFormat.Slides.dll








Represents the slide object within a presentation

```
public class Slide
```

## Inheritance

[object](#)  ← Slide

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

# Constructors

## Slide()

Constructor for the Slide class.

```
public Slide()
```

## Remarks

it initializes the Slide Facade set the slide index and initializes the lists of text shapes and images.

## Slide(bool)

```
public Slide(bool isNewSlide)
```

## Parameters

**isNewSlide** [bool](#) 

# Properties

## BackgroundColor

Property to set background color of a slide.

```
public string BackgroundColor { get; set; }
```

Property Value

[string](#)

## Images

Property contains the list of all images within a slide.

```
public List<Image> Images { get; set; }
```

Property Value

[List](#) <[Image](#)>

## RelationshipId

Property for the relationship Id.

```
public string RelationshipId { get; set; }
```

Property Value

[string](#)

## SlideFacade

Property for respective Slide Facade.

```
public SlideFacade SlideFacade { get; set; }
```

Property Value

[SlideFacade](#)

## SlideIndex

Property to hold the index of the slide.

```
public int SlideIndex { get; set; }
```

Property Value

[int](#)

## TextShapes

Property contains the list of all text shapes.

```
public List<TextShape> TextShapes { get; set; }
```

Property Value

[List](#) <[TextShape](#)>

## Methods

### AddImage(Image)

Method to add images to a slide.

```
public void AddImage(Image image)
```

Parameters

image [Image](#)

An object of Image class

## AddTextShapes(TextShape)

Method to add a text shape in a slide.

```
public void AddTextShapes(TextShape textShape)
```

Parameters

textShape [TextShape](#)

An object of TextShape class.

## AddTextShapes(TextShape, List<TextSegment>)

```
public void AddTextShapes(TextShape textShape, List<TextSegment> textSegments)
```

Parameters

textShape [TextShape](#)

textSegments [List](#) <[TextSegment](#)>

## GetTextShapesByText(string)

Get text shapes by searching a text term.

```
public List<TextShape> GetTextShapesByText(string text)
```

Parameters

text [string](#)

Search term as string

Returns

[List](#)  [<TextShape>](#)




# Class StyledList

Namespace: [FileFormat.Slides](#)








Assembly: FileFormat.Slides.dll

```
public class StyledList
```

## Inheritance

[object](#)  ← StyledList

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

## Constructors

### StyledList()

Constructor of StyledList class.

```
public StyledList()
```

## Properties

### Facade

Property to get the facade of a styled list

```
public ListFacade Facade { get; }
```

### Property Value

ListFacade

# ListItems

```
public List<string> ListItems { get; set; }
```

## Property Value

[List](#) <[string](#)>

## Methods

### AddListItem(string)

Method to add list items in styled list.

```
public void AddListItem(string text)
```

## Parameters

text [string](#)

It accepts text as list item

# Class TextSegment

Namespace: [FileFormat.Slides](#)

Assembly: FileFormat.Slides.dll








This class represents the text segment within a paragraph.

```
public class TextSegment
```

## Inheritance

[object](#)  ← TextSegment

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

## Properties

### Bold

Property to make bold the text segment.

```
public bool Bold { get; set; }
```

### Property Value

[bool](#) 

### Color

Property to set color the text segment.

```
public string Color { get; set; }
```

### Property Value

[string](#)

## Facade

Property to get facade of text segment.

```
public TextSegmentFacade Facade { get; }
```

Property Value

TextSegmentFacade

## FontFamily

Property to set font family.

```
public string FontFamily { get; set; }
```

Property Value

[string](#)

## FontSize

Property to set or get the font size of the text segment

```
public int FontSize { get; set; }
```

Property Value

[int](#)

## Italic

Property to make Italic the text segment.

```
public bool Italic { get; set; }
```

Property Value

[bool](#)↗

## Text

Property to set the text of the text segment.

```
public string Text { get; set; }
```

Property Value

[string](#)↗

## Methods

### create()

Method to create text segment.

```
public TextSegment create()
```

Returns

[TextSegment](#)

# Class TextShape

Namespace: [FileFormat.Slides](#)

Assembly: FileFormat.Slides.dll








This class represents the text shape within a slide.

```
public class TextShape
```

## Inheritance

[object](#)  ← TextShape

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

## Constructors

### TextShape()

Constructor of the TextShape class initializes the object of TextShapeFacade and populate its fields.

```
public TextShape()
```

## Properties

### Alignment

Property to get or set alignment of the shape.

```
public TextAlignment Alignment { get; set; }
```

### Property Value

[TextAlignment](#)

# BackgroundColor

Property to set or get background color of a text shape.

```
public string BackgroundColor { get; set; }
```

Property Value

[string](#) 

# Facade

Property to get or set the TextShapeFacade.

```
public TextShapeFacade Facade { get; set; }
```

Property Value

[TextShapeFacade](#)

# FontFamily

Property to get or set the font family of the text shape.

```
public string FontFamily { get; set; }
```

Property Value

[string](#) 

# FontSize

Property to set or get the font size of the Text Shape.

```
public int FontSize { get; set; }
```

Property Value

[int](#)

## Height

Property to get or set height of the shape.

```
public double Height { get; set; }
```

Property Value

[double](#)

## ShapeIndex

Property to get or set the shape index within a slide.

```
public int ShapeIndex { get; set; }
```

Property Value

[int](#)

## Text

Property to set or get the text of the shape.

```
public string Text { get; set; }
```

Property Value

[string](#)

## TextColor



Property to get or set the text color of the text shape.

```
public string TextColor { get; set; }
```

Property Value

[string](#) 

## TextList

Property to set or get styled list of a text shape.

```
public StyledList TextList { get; set; }
```

Property Value

[StyledList](#)

## TextSegments

Property to set or get text segments within a text shape.

```
public List<TextSegment> TextSegments { get; set; }
```

Property Value

[List](#)  [<TextSegment>](#)

## Width

Property to get or set width of the shape.

```
public double Width { get; set; }
```

Property Value

[double](#)

## X

Property to get or set X coordinate of the shape

```
public double X { get; set; }
```

Property Value

[double](#)

## Y

Property to get or set Y coordinate of the shape.

```
public double Y { get; set; }
```

Property Value

[double](#)

## Methods

### GetTextShapes(List<TextShapeFacade>)

Method for getting the list of text shapes.

```
public static List<TextShape> GetTextShapes(List<TextShapeFacade> textShapeFacades)
```

Parameters

**textShapeFacades** [List](#) <[TextShapeFacade](#)>

An object of TextShapeFacade.

Returns

[List](#) [<TextShape>](#)

## Remove()

Method to remove the textshape of a slide.

```
public void Remove()
```

## Update()

```
public void Update()
```

# Namespace FileFormat.Slides.Common

## Classes

### [Colors](#)

Common class to get the hexadecimal values of colors as string.

### [FormatException](#)

Custom exception class for file format-related exceptions.

### [Utility](#)

This class provides essential static methods for generating unique relationship IDs, obtaining random slide IDs, and converting measurements.

# Class Colors

Namespace: [FileFormat.Slides.Common](#)

Assembly: FileFormat.Slides.Common.dll








Common class to get the hexadecimal values of colors as string.

```
public static class Colors
```

## Inheritance

[object](#)  ← Colors

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

# Properties

## Aqua

Gets the hexadecimal value for the color Aqua (00FFFF).

```
public static string Aqua { get; }
```

## Property Value

[string](#) 

## Black

Gets the hexadecimal value for the color Black (000000).

```
public static string Black { get; }
```

## Property Value

[string](#)

## Blue

Gets the hexadecimal value for the color Blue (0000FF).

```
public static string Blue { get; }
```

Property Value

[string](#)

## Cyan

Gets the hexadecimal value for the color Cyan (00FFFF).

```
public static string Cyan { get; }
```

Property Value

[string](#)

## Fuchsia

Gets the hexadecimal value for the color Fuchsia (FF00FF).

```
public static string Fuchsia { get; }
```

Property Value

[string](#)

## Gray

Gets the hexadecimal value for the color Gray (808080).

```
public static string Gray { get; }
```

Property Value

[string](#) 

## Green

Gets the hexadecimal value for the color Green (00FF00).

```
public static string Green { get; }
```

Property Value

[string](#) 

## Lime

Gets the hexadecimal value for the color Lime (00FF00).

```
public static string Lime { get; }
```

Property Value

[string](#) 

## LimeGreen

Gets the hexadecimal value for the color Silver (C0C0C0).

```
public static string LimeGreen { get; }
```

Property Value

[string](#) 

# Magenta

Gets the hexadecimal value for the color Magenta (FF00FF).

```
public static string Magenta { get; }
```

Property Value

[string](#)

# Maroon

Gets the hexadecimal value for the color Maroon (800000).

```
public static string Maroon { get; }
```

Property Value

[string](#)

# Navy

Gets the hexadecimal value for the color Navy (000080).

```
public static string Navy { get; }
```

Property Value

[string](#)

# Olive

Gets the hexadecimal value for the color Olive (808000).

```
public static string Olive { get; }
```



Property Value

[string](#) 

## Orange

Gets the hexadecimal value for the color Orange (FFA500).

```
public static string Orange { get; }
```

Property Value

[string](#) 

## Purple

Gets the hexadecimal value for the color Purple (800080).

```
public static string Purple { get; }
```

Property Value

[string](#) 

## Red

Gets the hexadecimal value for the color Red (FF0000).

```
public static string Red { get; }
```

Property Value

[string](#) 

## Silver

Gets the hexadecimal value for the color Silver (C0C0C0).

```
public static string Silver { get; }
```

Property Value

[string](#) 

## Teal

Gets the hexadecimal value for the color Green (008000).

```
public static string Teal { get; }
```

Property Value

[string](#) 

## White

Gets the hexadecimal value for the color White (FFFFFF).

```
public static string White { get; }
```

Property Value

[string](#) 

## Yellow

Gets the hexadecimal value for the color Yellow (FFFF00).

```
public static string Yellow { get; }
```

Property Value

[string](#) 

# Class FileFormatException

Namespace: [FileFormat.Slides.Common](#)

Assembly: FileFormat.Slides.Common.dll

Custom exception class for file format-related exceptions.

```
public class FileFormatException : Exception, ISerializable
```

## Inheritance

[object](#) ← [Exception](#) ← FileFormatException

## Implements

[ISerializable](#)

## Inherited Members

[Exception.GetBaseException\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.GetType\(\)](#), [Exception.ToString\(\)](#), [Exception.Data](#), [Exception.HelpLink](#), [Exception.HResult](#), [Exception.InnerException](#), [Exception.Message](#), [Exception.Source](#), [Exception.StackTrace](#), [Exception.TargetSite](#), [Exception.SerializeObjectState](#), [object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#)

# Constructors

## FileFormatException(string, Exception)

Initializes a new instance of the [FileFormatException](#) class with a specified error message and a reference to the inner exception.

```
public FileFormatException(string message, Exception innerException)
```

## Parameters

**message** [string](#)

The error message that explains the reason for the exception.

**innerException** [Exception](#)

The exception that is the cause of the current exception, or a null reference if no inner exception is specified.

## Methods

### ConstructMessage(Exception, string)

```
public static string ConstructMessage(Exception Ex, string Operation)
```

#### Parameters

Ex [Exception](#) 

Operation [string](#) 

#### Returns

[string](#) 

# Class Utility

Namespace: [FileFormat.Slides.Common](#)

Assembly: FileFormat.Slides.Common.dll








This class provides essential static methods for generating unique relationship IDs, obtaining random slide IDs, and converting measurements.

```
public static class Utility
```

## Inheritance

[object](#)  ← Utility

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

# Properties

## NextIndex

Property to set next index for slide relationship Id.

```
public static int NextIndex { get; set; }
```

Property Value

[int](#) 

## SlideNextIndex

```
public static int SlideNextIndex { get; set; }
```

Property Value

[int](#) 

## Methods

### EmuToPixels(long)

Function to convert EMU to Pixel

```
public static double EmuToPixels(long emuValue)
```

#### Parameters

emuValue [long](#)

Long value

#### Returns

[double](#)

### GetRandomSlideId()

Function to get unique slide Id.

```
public static uint GetRandomSlideId()
```

#### Returns

[uint](#)

### GetUniqueRelationshipId()

Function to generate a unique Relationship ID

```
public static string GetUniqueRelationshipId()
```

#### Returns

[string](#)

## PixelsToEmu(double)

Function to convert Pixel value to EMU.

```
public static long PixelsToEmu(double pixelsValue)
```

### Parameters

**pixelsValue** [double](#)

Double value

### Returns

[long](#)



# Namespace FileFormat.Slides.Common. Enumerations

## Enums

### [TextAlignment](#)

Specifies the alignment of text elements.

# Enum TextAlignment

Namespace: [FileFormat.Slides.Common.Enumerations](#)

Assembly: FileFormat.Slides.Common.dll

Specifies the alignment of text elements.

```
public enum TextAlignment
```

## Fields

Center = 2

Left = 0

None = 3

Right = 1