PDF Optional Content Module

General about PDF Layer

Sample 1:how to dump optional content information of a PDF file

C#

```
String inputFilePath = "C:\\1.pdf";
String dumpFilePath = "C:\\dump.txt";

PDFOptionalContentHandler.Debug(inputFilePath, dumpFilePath);
```

Sample 2: how to retreive layer information of a PDF file

C#

```
String inputFilePath = "C:\\1.pdf";
String dumpFilePath = "C:\\dump.txt";

using (StreamWriter sw = newStreamWriter(inputFolder + "layers.txt", false))
{
List<PDFPageLayer> layers = PDFOptionalContentHandler.ExtractLayerInfos(inputFilePath);
foreach (PDFPageLayer layer in layers)
   {
      sw.WriteLine("Layer: {0} (ID:{1})", layer.Name, layer.ID);
      sw.WriteLine("Default State: {0}", (layer.DefaultState ? "ON" : "OFF"));
      sw.WriteLine("Property:");
      sw.WriteLine("CreateInfo:");
foreach (String key in layer.CreatorInfo.Keys)
      {
         sw.WriteLine("  {0}: {1}", key, layer.CreatorInfo[key]);
      }
   }
}
```

Sample 3: how to render a page layer of a PDF document

C#

```
//  To get all layer IDs in the document
List<int> layerIDs = PDFOptionalContentHandler.ExtractLayerIDs(inputFilePath);
foreach (int id in layerIDs)
{
Bitmap img = PDFOptionalContentHandler.RenderPageLayer(inputFilePath, 0, id);
   img.Save(inputFilePath + "output.layer" + id + ".png");
}
```

Import page layer

Sample 4: how to use PDFContext to create a resouce file for importing page layer

Create a resource with line content

C#

```
String resourceFilePath1 = "C:\\~resource1.pdf";
// Create a context with 2 inches * 2 inches
PDFContext ctx = newPDFContext(newSizeF(2F, 2F));
// Set resource background to transparent
ctx.SetBackGroundColor(Color.Transparent);
// Draw line
ctx.DrawLine(newREPen(Color.Red, 1F), 10, 10, 60, 60);
ctx.SaveToFile(resourceFilePath1);
```

Create a resource with outlines and shape content

C#

```
String resourceFilePath2 = "C:\\~resource2.pdf";
// Create a context with 4 inches * 3 inches
PDFContext ctx = newPDFContext(newSizeF(4F, 3F));
// Set resource background to transparent
ctx.SetBackGroundColor(Color.Transparent);
// Draw rectangle
ctx.DrawRectangle(newREPen(Color.Yellow, 1F), 10, 10, 50, 50);
// Fill ellipse
ctx.FillEllipse(newRESolidBrush(Color.Gray), 80, 80, 40, 30);
ctx.SaveToFile(resourceFilePath2);
```

Create a resource file with text content

C#

```
String resourceFilePath3 = "C:\\~resource3.pdf";
//  Create a context with 1 inches * 4 inches
PDFContext ctx = newPDFContext(newSizeF(1F, 4F));
//  Set resource background to transparent
ctx.SetBackGroundColor(Color.Transparent);
//  Draw text
REString text = REString.Create("Test Layer 1", newFont("Arial", 16F, FontStyle.Regular));
ctx.DrawString(newRESolidBrush(Color.Blue), 5, 5, text);
ctx.SaveToFile(resourceFilePath3);
```

Sample 5: how to create and import a new page layer to a PDF file

By file path

C#

```
String rootDir = @"C:\";
String inputFilePath = rootDir + "input.pdf";
String resourceFilePath = rootDir + "~resource1.pdf";
String outputFilePath = rootDir + "output.pdf";

//  Set new page layer's name to 'Test Layer 1'
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create("Test Layer 1");
//  Create an item from the resouce file.
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
//  Apply item to the 1st page with default settings.
item.TargetPageIndex = 0;
//  Import a new page layer to the input PDF file.
PDFOptionalContentHandler.ImportPageLayer(inputFilePath, outputFilePath, importArgs);
```

By file stream

C#

```
String rootDir = @"C:\";
String inputFilePath = rootDir + "input.pdf";
String resourceFilePath = rootDir + "~resource1.pdf";
String outputFilePath = rootDir + "output.pdf";

//  Set new page layer's name to 'Test Layer 1'
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create("Test Layer 1");
//  Create an item from the resouce file.
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
//  Apply item to the 1st page with default settings.
item.TargetPageIndex = 0;
using (FileStream inStream = File.Open(inputFilePath, FileMode.Open, FileAccess.Read,
FileShare.ReadWrite))
{
using (FileStream outStream = File.Open(outputFilePath, FileMode.Create, FileAccess.ReadWrite,
FileShare.ReadWrite))
    {
//  Import a new page layer to the input PDF file stream.
PDFOptionalContentHandler.ImportPageLayer(inStream, outStream, importArgs);
    }
}
```

Sample 6: how to import page layer items to an exist page layer in the PDF file

By file path

C#

```
String rootDir = @"C:\";
String inputFilePath = rootDir + "input.pdf";
String resourceFilePath = inputFolder + "~resource1.pdf";
String outputFilePath = inputFolder + "output.pdf";

//  Get all exist page layer IDs in the input PDF file.
List<int> existLayerIDs = PDFOptionalContentHandler.ExtractLayerIDs(inputFilePath);
//  Set the exist page layer with the specified ID.
int existLayerID = existLayerIDs[0];
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create(existLayerID);
//  Create an item from the resouce file.
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
//  Apply item to the 1st page with default settings.
item.TargetPageIndex = 0;
//  Import item to an exist page layer of the input PDF file.
PDFOptionalContentHandler.ImportPageLayer(inputFilePath, outputFilePath, importArgs);
```

By file stream

C#

```
String rootDir = @"C:\";
String inputFilePath = rootDir + "input.pdf";
String resourceFilePath = inputFolder + "~resource1.pdf";
String outputFilePath = inputFolder + "output.pdf";

//  Get all exist page layer IDs in the input PDF file.
List<int> existLayerIDs = PDFOptionalContentHandler.ExtractLayerIDs(inputFilePath);
//  Set the exist page layer with the specified ID.
int existLayerID = existLayerIDs[0];
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create(existLayerID);
//  Create an item from the resouce file.
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
//  Apply item to the 1st page with default settings.
item.TargetPageIndex = 0;
using (FileStream inStream = File.Open(inputFilePath, FileMode.Open, FileAccess.Read,
FileShare.ReadWrite))
{
using (FileStream outStream = File.Open(outputFilePath, FileMode.Create, FileAccess.ReadWrite,
FileShare.ReadWrite))
   {
//  Import item to an exist page layer of the input PDF file stream.
PDFOptionalContentHandler.ImportPageLayer(inStream, outStream, importArgs);
   }
}
```

Sample 7: how to batch import page layers to a PDF file

C#

```
String rootDir = @"C:\";
String inputFilePath = rootDir + "input.pdf";
String outputFilePath = inputFolder + "output.pdf";

String resourceFilePath1 = inputFolder + "~resource1.pdf";
String resourceFilePath2 = inputFolder + "~resource2.pdf";
String resourceFilePath3 = inputFolder + "~resource3.pdf";

//  Batch import page layers.
List<ImportPageLayerArgs> importArgsList = newList<ImportPageLayerArgs>();
{  //  1st
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create("Test Layer 1");
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath1);
   item.TargetPageIndex = 0;
   importArgsList.Add(importArgs);
}
{  //  2nd
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create("Test Layer 2");
ImportPageLayerArgs.Item item1 = importArgs.AddItem(resourceFilePath2);
   item1.TargetPageIndex = 0;
ImportPageLayerArgs.Item item2 = importArgs.AddItem(resourceFilePath3);
   item2.TargetPageIndex = 1;
   importArgsList.Add(importArgs);
}
{  //  3rd
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create("Test Layer 3");
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath3);
   item.TargetPageIndex = 0;
   importArgsList.Add(importArgs);
}
//  Batch import items to page layers of the input PDF file.
PDFOptionalContentHandler.ImportPageLayer(inputFilePath, outputFilePath, importArgsList.ToArray());
```

Use import page layer argument

Sample 8: how to create an ImprotPageLayerArgs

For importing items to a new page layer

C#

```
...
//  Import item(s) to a new page layer.
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create("Test Layer 1");
...
```

For importing items to an exist page layer

C#

```
...
//  This must be a valid page layer ID in the target PDF file.
int existPageLayerID = 11;
//  Import item(s) to a new page layer.
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create(existPageLayerID);
...
```

Sample 9: how to add an importing item to ImprotPageLayerArgs

Create an item form a PDF resource file

C#

```
String resourceFilePath = inputFolder + "~resource1.pdf";
...
//  Import item(s) to a new page layer.
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create("Test Layer 1");
//  Add an item which created from a PDF file's first page.
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath, 0);
...
```

Create an item from a System.Drawing.Bitmap object

C#

```
String resourceFilePath = inputFolder + "~resource1.png";
Bitmap resImage = newBitmap(resourceFilePath);
...
//  Import item(s) to a new page layer.
ImportPageLayerArgs importArgs = ImportPageLayerArgs.Create("Test Layer 1");
//  Add an item which created from a System.Drawing.Bitmap
ImportPageLayerArgs.Item item = importArgs.AddItem(resImage);
...
```

Sample 10: how to set target page for an importing item

C#

```
...
//  Add an item which created from a PDF file's first page.
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath, 0);
//  This item would be added to the 1st page.
item.TargetPageIndex = 0;
...
```

Sample 11: how to set the importing item's size in the target page

Use the original size of the resource

C#

```
...
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
...
item.IsAbsolute = true;
item.ScaleWRatioInPercent = 100;    //  100%
item.ScaleHRatioInPercent = 100;    //  100%
...
```

Use 50% of the original size of the resource

C#

```
...
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
...
item.IsAbsolute = true;
item.ScaleWRatioInPercent = 50;    //  50%
item.ScaleHRatioInPercent = 50;    //  50%
...
```

Use the size of the target page for the importing item

C#

```
...
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
...
item.IsAbsolute = false;
item.ScaleWRatioInPercent = 100;    //  100%
item.ScaleHRatioInPercent = 100;    //  100%
...
```

Use 50% of the target page for the importing item

C#

```
...
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
...
item.IsAbsolute = false;
item.ScaleWRatioInPercent = 50;    //  50%
item.ScaleHRatioInPercent = 50;    //  50%
...
```

Sample 12: how to set the importing item’s location in the target page

Put the importing item at the center of the target page

C#

```
…
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
…
item.Alignment = ContentAlignment.MiddleCenter;
…
```

Align the importing item to the top left corner of the target page without shifting

C#

```
…
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
…
item.Alignment = ContentAlignment.TopLeft;
…
```

Align the importing item to the top left corner of the target page with shifting

C#

```
…
ImportPageLayerArgs.Item item = importArgs.AddItem(resourceFilePath);
…
item.Alignment = ContentAlignment.TopLeft;
//  Shift item 2 inches (192 pixels) to left and 1 inch (96 pixels) to bottom.
//  Unit: pixel (96 dpi)
item.ShiftX = 192;
item.ShiftY = 96;
…
```

Merge Page Layers

Sample 13: how to merge page layers in a PDF file

For file

C#

```
String inputFolder = @"C:\";
String inputFilePath = inputFolder + "input.pdf";
String outputFilePath = inputFolder + "output.pdf";

//  Get all exist page layer IDs in the input PDF file.
List<int> existPageLayers = PDFOptionalContentHandler.ExtractLayerIDs(inputFilePath);
if (existPageLayers.Count < 2)
{
Console.WriteLine("No enough page layers in the input PDF file for this testing!");
return;
}

//  Merge all page layers to the last page layer.
List<uint> srcIDs = newList<uint>();
for (int i = 0; i < existPageLayers.Count - 1; i++)
    srcIDs.Add((uint)existPageLayers[i]);
uint destID = (uint)existPageLayers[existPageLayers.Count - 1];

//  Merging some page layers to a page layer.
PDFOptionalContentHandler.MergePageLayers(inputFilePath, outputFilePath, srcIDs.ToArray(), destID);
```

For file stream

C#

```
String inputFolder = @"C:\";
String inputFilePath = inputFolder + "input.pdf";
String outputFilePath = inputFolder + "output.pdf";

//  Get all exist page layer IDs in the input PDF file.
List<int> existPageLayers = PDFOptionalContentHandler.ExtractLayerIDs(inputFilePath);
if (existPageLayers.Count < 2)
{
Console.WriteLine("No enough page layers in the input PDF file for this testing!");
return;
}

//  Merge all page layers to the last page layer.
```

```
List<uint> srcIDs = newList<uint>();
for (int i = 0; i < existPageLayers.Count - 1; i++)
   srcIDs.Add((uint)existPageLayers[i]);
uint destID = (uint)existPageLayers[existPageLayers.Count - 1];

using (FileStream inStream = File.Open(inputFilePath, FileMode.Open, FileAccess.Read,
FileShare.ReadWrite))
{
using (FileStream outStream = File.Open(outputFilePath, FileMode.Create, FileAccess.ReadWrite,
FileShare.ReadWrite))
   {
//  Merging some page layers to a page layer.
PDFOptionalContentHandler.MergePageLayers(inStream, outStream, srcIDs.ToArray(), destID);
   }
}
```

Flatten Page Layers

Sample 14: how to flatten all page layers in a PDF file

For file

C#

```
String inputFolder = @"C:\";
String inputFilePath = inputFolder + "input.pdf";
String outputFilePath = inputFolder + "output.pdf";

//  Flatten all page layers in the PDF file.
PDFOptionalContentHandler.FlattenPageLayers(inputFilePath, outputFilePath);
```

For file stream

C#

```
String inputFolder = @"C:\";
String inputFilePath = inputFolder + "input.pdf";
String outputFilePath = inputFolder + "output.pdf";

using (FileStream inStream = File.Open(inputFilePath, FileMode.Open, FileAccess.Read,
FileShare.ReadWrite))
{
using (FileStream outStream = File.Open(outputFilePath, FileMode.Create, FileAccess.ReadWrite,
FileShare.ReadWrite))
    {
//  Flatten all page layers in the PDF file.
PDFOptionalContentHandler.FlattenPageLayers(inStream, outStream);
    }
}
```