##### Example: TutorialC

# Purpose

TutorialC project is an example of а programming tutorial book. The example demonstrates creation of a multi-page text document with mixed content. The document includes several sections, and a repeating footer with automatic page numeration. Additionally, this example shows how to use images and how to work with tables.

# Prerequisites

1) **Visual Studio 2017 or above is installed. To install a community version of Visual Studio use the following link:** [**https://visualstudio.microsoft.com/vs/community/**](https://visualstudio.microsoft.com/vs/community/) **Please make sure that the way you are going to use Visual Studio is allowed by the community license. You may need to buy Standard or Professional Edition.**

2) **.NET Core SDK 2.1 or above is installed. To install the framework use the following link:** [**https://dotnet.microsoft.com/download**](https://dotnet.microsoft.com/download)

# Description

### Image files

The document image files is located in the folder **Images\TutorialC.**

### Additional data

The document additional data is located in the file **Content/tutorialc\_keywords.json. The file contains a description of the C keywords. This data is used to build the table. The file is a standard JSON.**

[  
{  
   "Id": "1",  
   "Keyword": "auto",  
   "Description": "The auto keyword declares automatic variables"  
},  
{  
   "Id": "2",  
   "Keyword": "break",  
   "Description": "The break statement makes program jump out of the innermost enclosing loop (while, do, for or switch statements) explicitly."  
},  
​  
]  
//etc.

### Output file

The example creates the file **TutorialC.pdf in the output bin/(debug|release)/netcoreapp2.1 folder.**

# Writing the source code

## Create project

#### 1. Create new console application.

1.1. Run Visual Studio 1.2. File -> Create -> Console Application (.Net Core)

#### 2. Modify class Program.

2.1. In the function **Main() set path to output the PDF file:**

   var pathDocument = "TutorialC.pdf";

2.2. Call method **Run() for resume generation and Build() for building a document into output PDF:**

   TutorialCRunner.Start(pathDocument);

2.3. After generation is completed, notify user of successful generation:

   Console.WriteLine("document has been successfully built");

## Sources structure

The document consists of a few logical parts, and we use Create methods to generate each logical part:

* AddBookCoverSection() - create the cover image of the book
* AddFirstSection() - create the first section of the book
* AddSecondSection() - create the 'Basic syntax' section of the book
* AddThirdSection() - create the 'Desicion making' section of the book
* AddOutlineSection() - add an outline (Table of Contents) to the book

## Global definitions (TutorialCRunner() constructor)

Let's define the necessary paths.

ProjectDir = Directory.GetCurrentDirectory();

Let's define the required data from the JSON data file.

KeywordsJsonFile = Path.Combine(ProjectDir, "Content", "tutorialc\_keywords.json");  
KeywordsJsonContent = File.ReadAllText(KeywordsJsonFile);  
KeywordsList = JsonConvert.DeserializeObject<List<Keywords>>(KeywordsJsonContent);

Let's define the path for the image files.

ImageUrl = Path.Combine(ProjectDir, "images", "TutorialC", "customers");

Let's define the fonts.

DocumentFont = Fonts.Times(12f);  
//etc.

## Optimization of the code.

To add footers several times with a different text, we added a new method. Defining methods for common operations is a good optimization practice to achieve better code size.

   internal static void AddFooters(SectionBuilder section, string text)  
  {  
       var imageUrlLogo = Path.Combine(ProjectDir, "images", "DocumentExample", "LearnCLogo30\_30.jpg");  
       section.AddFooterToBothPages(40)  
          .AddLine()  
              .SetColor(ColorBackground).SetStroke(Stroke.Solid).SetWidth(1f)  
          .ToArea()  
              .AddParagraph()  
                  .SetMargins(0, 5, 0, 0).SetFont(DocumentFontBoldOrange)  
                  .AddInlineImageToParagraph(imageUrlLogo,  
                       new XSize(15, 15), ScalingMode.UserDefined)  
                  .AddTextToParagraph(text)  
                  .AddUrl("http://www.gehtsoftusa.com/", "GEHTSOFT USA LLC.")  
                      .SetFont(DocumentFontBoldOrange)  
                      .SetUnderline(ColorBackground);  
       section.AddFooterToBothPages(15f)  
          .AddParagraph()  
              .AddPageNumberToParagraph(" Page #", 1, SmallFont)  
              .SetAlignment(HorizontalAlignment.Right);  
  }

## Cover section (AddBookCoverSection() method)

We used an image for the cover of the book. Let's define the path to the image file.

var imageUrl = Path.Combine(ProjectDir, "images", "TutorialC", "DocumentExample", "LearnCCoverPage.jpg");

We also want to add a link from the table of contents. To do this, let's define a hidden font.

var fontHidden = Fonts.Courier(.01f).SetColor(Color.White);

Let's add a hidden label for the table of contents.

builder  
  .AddSection()  
      .SetSize(PaperSize.A4)  
      .SetOrientation(PageOrientation.Portrait)  
      .AddParagraph("LEARN C PROGRAMMING")  
          .SetFont(fontHidden)  
          .SetBackColor(Color.White).SetOutline().SetOutline(0)

Let's add an image to decorate the book cover.

  .ToSection()  
      .AddImage(imageUrl).SetScale(ScalingMode.Stretch);

## First section 'Environment setup' (AddFirstSection() method)

Let's define the necessary paths to the images: imageUrlLogo, imageUrl, imageExclamationMark, imageQuestionMark. Let's create a title that will be added to the table of contents.

  .AddParagraph()  
      .SetFont(DocumentFontTitleWhite)  
      .SetAlignment(HorizontalAlignment.Center)  
      .SetBackColor(ColorBackground)  
      .SetBorder(Stroke.Solid, Color.Gray, 3)  
      .SetOutline(1)  
      .AddText("1. Environment setup")

Next we will add paragraphs and content to the section. We add images to paragraphs so that the reader's attention is drawn to the important information.

  .AddInlineImageToParagraph(imageExclamationMark, new XSize(15, 15), ScalingMode.UserDefined)

If you need to skip line breaks in the text, you can specify the ignoreNewLineSymbol parameter:

  .AddText(@" to set up your own environment to start learning C   
programming language. Reason is very simple, we already have set up C   
Programming environment online, so that you can compile and execute all the   
available examples online at the same time when you are doing your theory   
work. This gives you confidence in what you are reading and to check the result   
with different options. Feel free to modify any example and execute it online.",  
           ignoreNewLineSymbol: true)

Let's add a hyperlink to indicate the useful source of information.

  .AddUrlToParagraph("http://www.compileonline.com/")

To accurately convey the text of the program, we used the @ strings and passed the strings as an array.

  .AddText(  
   new[]  
  {  
       "#include <stdio.h>",  
       " ",  
       "int main()",  
       "{",  
       @"   /\* my first program in C \*/",  
       @"   printf(""Hello, World! \n"");",  
       @"   return 0;",  
       "}"  
  })

## Second section 'Basic syntax' (AddSecondSection() method)

Let's define the necessary paths to the images: imageUrlLogo, imageExclamationMark. Let's create a title that will be added to the table of contents.

  .AddParagraph("2. Basic syntax").SetFont(DocumentFontTitleWhite)  
      .SetAlignment(HorizontalAlignment.Center)  
      .SetBackColor(ColorBackground)  
      .SetBorder(Stroke.Solid, Color.Gray, 3)  
      .SetOutline(1)

Add paragraphs and content to the section.

To accurately convey the text of the bitwise operators, we used strings as an array.

  .AddText(  
       new[]  
      {  
           "A = 0011 1100", "B = 0000 1101", "A&B = 0000 1100", "A|B = 0011 1101", "A^B = 0011 0001",  
           "~A = 1100 0011"  
      })

We have added a footer to this section to show the title of the chapter.

   AddFooters(s, " C tutorial. Chapter #2. Basic syntax. ");

## Third section 'Desicion making' (AddThirdSection() method)

Let's define the necessary paths to the images: imageUrlLogo, imageUrl. Let's create a title that will be added to the table of contents.

  .AddParagraph()  
          .AddTextToParagraph("3. Desicion making")  
          .SetFont(DocumentFontTitleWhite)  
          .SetAlignment(HorizontalAlignment.Center)  
          .SetBackColor(ColorBackground)  
          .SetBorder(Stroke.Solid, Color.Gray, 3)  
          .SetOutline(1)

Add paragraphs and content to the section.

If you need to skip line breaks in the text, you can specify the ignoreNewLineSymbol parameter:

  .AddTextToParagraph(@"Decision-making structures require that the programmer specifies one or more   
conditions to be evaluated or tested by the program, along with a statement or   
statements to be executed if the condition is determined to be true, and   
optionally, other statements to be executed if the condition is determined to be   
false.", ignoreNewLineSymbol: true)

The **fluent style allows us to switch after the paragraph to the section context (ToSection). After that, we add an image.**

.ToSection()  
  .AddImage(imageUrl)  
      .SetScale(ScalingMode.OriginalSize)  
      .SetAlignment(HorizontalAlignment.Center)  
      .SetBorder(Stroke.Dashed, ColorBackground, 3)

We have added a footer to this section to show the title of the chapter.

   AddFooters(s, " C tutorial. Chapter #3. Decision making. ");

## Outline section (AddOutlineSection() method)

We have added a title to the page.

  .AddParagraph()  
      .AddTextToParagraph("CONTENTS")  
      .SetFont(DocumentFontTitleWhite)  
      .SetAlignment(HorizontalAlignment.Center)  
      .SetBackColor(ColorBackground)  
      .SetBorder(Stroke.Solid, Color.Gray, 3)

To add a table of contents, just call the appropriate method: builder.AddOutline(). You can add parameters.

   builder.AddOutline()  
          .SetSpacingUnderline(Stroke.Dashed, ColorBackground)  
          .SetLevelLeftIndent(10f)  
          .SetFont(DocumentFontBoldOrange);

We have added a footer to this section to show the title.

   AddFooters(s, " C tutorial. Contents. ");

# Summary

The reader has learned how to make a real reference book. The article explains how to add a table of contents structure. This article explains how to add images to a reference book. This article explains how to add tables with data from a file JSON.