How to read barcodes from subfolder with barcode reader sdk in Delphi and ByteScout Premium Suite

Continuous learning is a crucial part of computer science and this tutorial shows how to read barcodes from subfolder with barcode reader sdk in Delphi

Source code documentation samples give simple and easy method to install a needed feature into your application. ByteScout Premium Suite is the bundle that includes twelve SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording. It can be applied to read barcodes from subfolder with barcode reader sdk using Delphi.

Want to save time? You will save a lot of time on writing and testing code as you may just take the Delphi code from ByteScout Premium Suite for read barcodes from subfolder with barcode reader sdk below and use it in your application. Follow the instructions from scratch to work and copy the Delphi code. Enjoy writing a code with ready-to-use sample Delphi codes.

ByteScout Premium Suite free trial version is available on our website. Delphi and other programming languages are supported.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Premium Suite

Explore API Documentation

Get Free Training for ByteScout Premium Suite

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

Project1.dpr

```pascal
//*********************************************************************
//       ByteScout Barcode Reader SDK
//
//       Copyright � 2016 ByteScout - http://www.bytescout.com
//       ALL RIGHTS RESERVED
//
//*********************************************************************

program Project1;

{$APPTYPE CONSOLE}

{
 Sample that shows reading of barcodes from images in /Images/ subfolder
}

uses
  SysUtils,
  ComObj,
  ActiveX;
var
 reader: Variant;
 SR: TSearchRec;
 arrayOutput: Variant;
 varItem: Variant;
 i: integer;
 pathWithImages, imageType: string;
begin
 CoInitialize(nil);

 // Disable floating point exception to conform to .NET floating point operations beha
 System.Set8087CW(

{codeFileName}




{code}




33f);

 // Create and initialize Bytescout.BarCodeReader.Reader object
 reader := CreateOleObject('Bytescout.BarCodeReader.Reader');
 reader.RegistrationName := 'demo';
 reader.RegistrationKey := 'demo';
```

```delphi
  // Enable Code 39 decoding, you may also enable other types (dozens of types supported
  // see .BarCodeTypesToFind for more values.
  reader.BarcodeTypesToFind.Code39 := true;

  // For example to enable Code128 uncomment line below
  // reader.BarcodeTypesToFind.Code128 := true;

  // For example to enable QR Code uncomment line below
  // reader.BarcodeTypesToFind.QRCode := true;

  // Setup possible barcode orientations ( ato support rotated images)
  // see BarCode Reader SDK - OrientationType enum for more variations
  // uncomment to support more orientations like right to left
  {
  reader.Orientation :=
                                  1 or //OrientationType.HorizontalFromLeftToRigh
                                  16 or // OrientationType.HorizontalFromRightToL
                                  32768 or // OrientationType.VerticalFromBottomT
                                  2; // OrientationType.VerticalFromTopToBottom;
  }

  // uncomment to enable processing of negative barcodes (white bars on dark background]
  // reader.SearchNegative := true;

  // uncomment to Enable Heuristic Mode (try different image processing modes if the de
  // Helps on photos with poor lighting conditions.
  //reader.HeuristicMode := true;

  // define subfolder where we have images
  pathWithImages := '.\Images\';
  imageType := '*.jpg'; // can also be *.png, *.tiff, *.bmp, *.PDF

  // now search for files with images and processing them one by one
  if FindFirst(pathWithImages + imageType, faAnyFile, SR) = 0 then begin
    repeat
      if (SR.Attr <> faDirectory) then begin
        writeLn(SR.Name);
        writeLn('------------------');
        // processing the file
        reader.ReadFromFile(pathWithImages + SR.Name);
        for i := 0 to reader.FoundCount - 1 do begin
          writeLn(
            '#' + IntToStr(i) + ' barcode with value ' +
            reader.GetFoundBarcodeValue(i)+
            ' at ' + FloatToStr(reader.GetFoundBarCodeLeft(i)) +
            ',' + FloatToStr(reader.GetFoundBarCodeTop(i))
          );
        end;
        writeLn('------------------');
      end;
    until FindNext(SR) <> 0;
    FindClose(SR);
  end;

  WriteLn('Press any key to exit...');
  ReadLn;

  reader := varEmpty;

end.
```

VIDEO

https://www.youtube.com/watch?v=NEwNs2b9YN8

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Premium Suite Home Page
Explore ByteScout Premium Suite Documentation
Explore Samples
Sign Up for ByteScout Premium Suite Online Training

ON-DEMAND REST WEB API

Get Your API Key
Explore Web API Docs
Explore Web API Samples

visit www.ByteScout.com

visit www.PDF.co