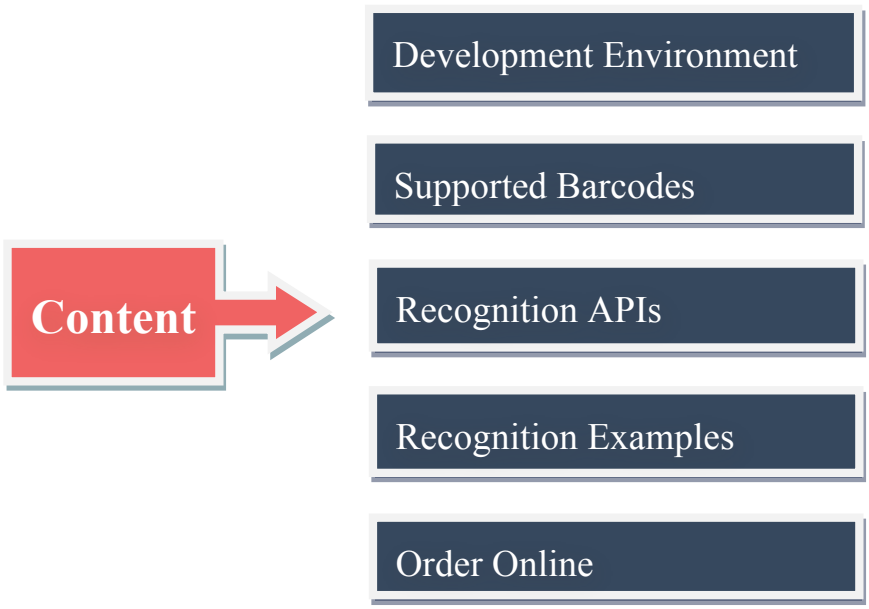
User Manual

CnetSDK.Barcode.Scanner.Trial.dll

Version: 17.5.0.1

Date: 08th, Sep 2017


Content
Development Environment
Supported Barcodes
Recognition APIs
Recognition Examples
Order Online

Development Environment

- Windows XP or Above, and Windows Server 2003 or Above
- 32 or 64 Bit System
- Visual Studio 2005 or Above
- .NET Framework 2.0 or Above

Supported Barcodes

1D/Linear Barcodes:

- Codabar
- Code39
- Code93
- Code128
- EAN8
- EAN13
- ITF14
- UPCA
- UPCE

2D/Matrix Barcodes:

- Aztec
- PDF417
- QRCode
- DataMatrix

Recognition APIs

CnetSDK .NET Barcode Scanner SDK supports reading most popular 1D and 2D barcodes from various image file formats in .NET application, including bmp, jpg, jpeg, png, tif, and gif. In all, nine APIs are included for high quality barcode recognition from image stream, bitmap and local image file.

To begin with, please firstly add CnetSDK.Barcode.Scanner.Trial.dll to your Visual Studio .NET project reference.

1. APIs for Scanning and Reading All Barcodes from Image

```
public static ScanResult[] ScanBarcode(Bitmap Bitmap);
public static ScanResult[] ScanBarcode(Stream Stream);
public static ScanResult[] ScanBarcode(string FileName);
```

2. APIs for Scanning and Reading A Specific Barcode Type from Image

```
public static ScanResult[] ScanBarcode(Bitmap Bitmap, CSBarcodeType Type);
public static ScanResult[] ScanBarcode(Stream Stream, CSBarcodeType Type);
public static ScanResult[] ScanBarcode(string FileName, CSBarcodeType Type);
```

3. APIs for Scanning and Reading The Only Barcode from Image

```
public static ScanResult[] ScanOnlyBarcode(Bitmap Bitmap, CSBarcodeType Type);
public static ScanResult[] ScanOnlyBarcode(Stream Stream, CSBarcodeType Type);
public static ScanResult[] ScanOnlyBarcode(string FileName, CSBarcodeType Type);
```

Please Note:

If you are using free trial package, the first character of barcode data will be recognized as CnetSDK*.

Recognition Examples

The following part illustrates several snipped C# demo codes for recognizing all barcodes, specific barcode type, and the only one barcode from image stream, bitmap and local image file. Please firstly integrate CnetSDK.Barcode.Scanner.Trial.dll into your VS .NET project by adding reference.

1. Scan All Barcodes

(1) Scan All Barcodes from **Image Stream**

```
public void ScanAllBarcodesfromStream(Stream Stream)
    {
      ScanResult[] results = CSBarcodeScanner.ScanBarcode(Stream);
      foreach (ScanResult output in results)
       {
         Console.WriteLine(output.BarcodeType.ToString() + "-" + output.BarcodeData);
       }
            Console.ReadKey();
    }
```

(2) Scan All Barcodes from **Bitmap**

```
public void ScanAllBarcodesfromBitmap(Bitmap Bmp)
    {
      ScanResult[] results = CSBarcodeScanner.ScanBarcode(Bmp);
      foreach (ScanResult output in results)
       {
         Console.WriteLine(output.BarcodeType.ToString() + "-" + output.BarcodeData);
       }
            Console.ReadKey();
    }
```

(3) Scan All Barcodes from **Local Image File**

```
public void ScanAllBarcodesfromFile(string FileName)
    {
      ScanResult[] results = CSBarcodeScanner.ScanBarcode(FileName);
      foreach (ScanResult output in results)
      {
        Console.WriteLine(output.BarcodeType.ToString() + "-" + output.BarcodeData);
      }
            Console.ReadKey();
    }
```

2. Scan A Specific Barcode Type

(1) Scan Code 39 from **Image Stream**

```
public void ScanCode39fromStream(Stream Stream)
    {
      ScanResult[] results = CSBarcodeScanner.ScanBarcode(Stream,
CSBarcodeType.Code39);
      foreach (ScanResult output in results)
      {
        Console.WriteLine(output.BarcodeType.ToString() + "-" + output.BarcodeData);
      }
            Console.ReadKey();
    }
```

(2) Scan Code 39 from **Bitmap**

```
public void ScanCode39fromBitmap(Bitmap Bmp)
    {
      ScanResult[] results = CSBarcodeScanner.ScanBarcode(Bmp, CSBarcodeType.Code39);
      foreach (ScanResult output in results)
      {
        Console.WriteLine(output.BarcodeType.ToString() + "-" + output.BarcodeData);
      }
```

```
            Console.ReadKey();
    }
```

(3) Scan Code 39 from **Local Image File**

```
  public void ScanCode39fromFile(string FileName)
    {
      ScanResult[] results = CSBarcodeScanner.ScanBarcode(FileName,
CSBarcodeType.Code39);
      foreach (ScanResult output in results)
      {
        Console.WriteLine(output.BarcodeType.ToString() + "-" + output.BarcodeData);
      }
            Console.ReadKey();
    }
```

To read other barcode types from image, please directly change barcode type to a specific one. All supported barcode types are listed in **Supported Barcode** section.

3. Scan The Only Barcode

This method is suitable when there is only one barcode on your image. It will improve the accuracy of barcode recognition.

```
  public void ReadtheOnlyBarcodefromImage(Bitmap Bmp)
    {
        Bitmap Bmp = new Bitmap("CnetSDKTest.jpg");
        ScanResult ReadBarcode = CSBarcodeScanner.ScanOnlyBarcode(Bmp);
        Console.WriteLine("barcode data:{0}.", ReadBarcode.BarcodeData);
        Console.ReadKey();
    }
```

Order Online

If you want to use CnetSDK product for commercial use, please order respective product license. After your payment is confirmed, you will receive an email including a download link for CnetSDK product (full product package).

All CnetSDK products are cost-effective and competitive on the market. Please see licensing details (license type, price, software license agreement) from online page: http://www.cnetsdk.com/order.

If you have any questions, please send email to support@cnetsdk.com.