



QS-Barcode SDK (32-Bit/64-Bit)

Beschreibung der Programmbeispiele

Mit dem **QS-Barcode SDK** (Software Development Kit) können Funktionen zur Barcodeerkennung leicht in Windows-Applikationen eingebunden werden. Dieses Papier ergänzt die QS-Barcode SDK Dokumentation. Übersichtlich werden die Programmbeispiele dargestellt, die beim SDK mitgeliefert werden. Die Gruppierung erfolgt nach Art der Schnittstelle (F_DLL, F_OCX, H_DLL, P_LIB) und nach Programmiersprache. Eine Diskussion der unterschiedlichen Schnittstellen befindet sich in der SDK Dokumentation. Für Details zu den Beispielprogrammen schauen Sie bitte auch in die readme-Dateien und in die Kommentare der Programmbeispiele.

Die in diesem Dokument beschriebenen Programmbeispiele zum QS-Barcode SDK erhalten Sie zusammen mit dem **QS-Barcode SDK**, dessen Evaluationsversion Sie kostenlos von unserer Homepage www.qualitysoft.de downloaden können. Nach der Ausführung des Setups stehen die Dateien im Installationsverzeichnis zur Verfügung.

Einleitung

Alle Beispiele liegen, soweit es die Programmiersprache unterstützt, als 32-Bit und 64-Bit Versionen vor und enthalten den Source-Code und die Projektdateien. Da der Aufruf der Funktionen zum Setzen der Barcode-Parameter und zum Lesen der Barcodes sehr einfach ist, können Sie auch leicht bei anderen Programmiersprachen abgucken, wenn das gewünschte Beispiel für Ihre Sprache nicht zur Verfügung steht. Wir wissen von unseren Kunden, dass QS-Barcode SDK auch für weitere Sprachen integriert wurde.

Bei Fragen oder Hinweisen senden Sie uns einfach eine E-Mail, wir nehmen dann schnell mit Ihnen Kontakt auf.

Hinweis: Die Evaluationsversionen verfälschen die Ergebnisse, aus 3 wird 1, aus A wird Q, aus B wird S. Bei 2D Barcodes erscheint sporadisch der Text „DEMO“ im Ergebnis. Wenn Sie die Lizenzdatei bestellen und diese in Ihr Applikationsverzeichnis kopieren, haben Sie ohne weitere Programmierarbeit die Vollversion.

Verzeichnis F_DLL(x86/x64)\sample (File - Schnittstelle)

Die Variante QSBarDLL_F.dll aus dem QS-Barcode SDK wird eingesetzt, wenn QS-Barcode SDK das Laden der Bilddatei eigenständig durchführen soll.

Die angegebene Datei wird geöffnet und mit den übergebenen Barcodeparametern wird die Barcodesuche und -lesung durchgeführt.

Hinweise!

Dateiformate

Die Beispielprogramme zur File-Schnittstelle unterstützen, sofern sie nicht mit einem festen Dateinamen arbeiten, viele Dateiformate. Eine komplette Liste finden in der QS-Barcode SDK Dokumentation im Anhang „15.1 Unterstützte Dateiformate“

MultiPage-Dateien

Bei mehrseitigen Dateien (MultiTiff oder Adobe PDF) wird normalerweise nur auf der ersten Seite nach Barcodes gesucht. Beachten Sie den Anhang „15.2 MultiPage Unterstützung“ aus der QS-Barcode SDK Dokumentation für Hinweise, wie Sie auch die weiteren Seiten durchsuchen können.

Adobe PDF Dokumente

Das Adobe PDF Format wird vom QS-Barcode SDK ab Version 3.8 unterstützt.

Beachten Sie den Anhang „15.3 Adobe PDF Dokumente – Spezielle Einstellungen“ aus der QS-Barcode SDK Dokumentation.



Unterverzeichnis	Sprache / Umgebung	Aufgabe
X86\Acc00	MS Access 2000, VBA	<p>Startet man die Access 2000 Datenbank, zeigt sie ein einfaches Formular mit einer Schaltfläche. Diese startet die Barcodeerkennung auf dem beiliegenden Bild <code>sample3.tif</code>. Es können lineare Barcodes, PDF417 und DataMatrix gelesen werden, die jeweiligen Ergebnisse werden in einer Messagebox angezeigt.</p> <p>Mit F11 gelangt man in die Datenbank und kann sich im Module <code>mdlBarcode</code> die Funktion <code>BarcodeRecognition()</code> ansehen. Dort wird der Aufruf der DLL aus Access heraus demonstriert. Die Barcodeparameter sind fest codiert, ebenso wie der benutzte Dateiname.</p>
X86\AccXP	Access XP	wie oben
X86\c\Default X64\c\Default	C	<p>Das Programm erwartet als Kommandozeilenparameter einen Dateinamen auf ein Bild. Die Bilddatei wird nach linearen Code 39 und Code128 Barcodes durchsucht und die gefundenen Barcodes auf der Standardausgabe ausgegeben. Die Barcodeparameter sind fest im Code eingestellt. Der Code zum Lesen von 2D Barcodes (PDF417, DataMatrix, QR Code oder Aztec Code) ist in der Quelle (auskommentiert) vorhanden. Das Beispielprogramm wurde mit MS Visual Studio 2008 geschrieben und getestet.</p>
X86\c\Whitepixel X64\c\Whitepixel	C	<p>Die Bilddatei nach wird nach einem Interleaved 2/5 Barcode durchsucht. Der Funktion zum Suchen von Barcodes wird dabei eine spezifische Konfigurationsdatei (<code>qsbc.ini</code>) zur Bildverbesserung übergeben, die bestehende Fehler auf dem Bild vor der Suche korrigiert. Der gefundene Barcode wird auf der Standardausgabe ausgegeben. Die Barcodeparameter sind fest im Code eingestellt. Das Beispielprogramm wurde mit MS Visual Studio 2008 geschrieben und getestet.</p>
X86\c\AdvancedSearch X64\c\AdvancedSearch	C	<p>Der Barcode kann im Standardfalle nicht gelesen werden. Vor der Lesung werden Reparaturen auf dem Barcode ausgeführt und der Barcode wird mit erweiterter Funktionalität gelesen.</p>
X86\c\DynamicThreshold X64\c\DynamicThreshold	C	<p>Die Bilddatei ist insgesamt zu dunkel für den Standard-Schwellwert (160). Daher erfolgt vor der Lesung eine dynamische Ermittlung des Schwellwertes, der das Bild in einen lesbaren Zustand versetzt.</p>
X86\c_sharp\default X64\c_sharp\default	C#	<p>Programm zur Eingabe eines Dateinamens. Aus dieser Datei werden lineare (oder, nach Entfernung von Programmkommentaren, PDF417 bzw. DataMatrix) Barcodes gelesen. Die Ergebnisse werden in einer Messagebox angezeigt</p>
X86\c_sharp\ AdvancedSearch X64\c_sharp\ AdvancedSearch	C#	<p>Der Barcode kann im Standardfalle nicht gelesen werden. Vor der Lesung werden Reparaturen auf dem Barcode ausgeführt und der Barcode wird mit erweiterter Funktionalität gelesen.</p>



X86\c_sharp\ DynamicThreshold X64\c_sharp\ DynamicThreshold	C#	Die Bilddatei ist insgesamt zu dunkel für den Standard-Schwelwert (160). Daher erfolgt vor der Lesung eine dynamische Ermittlung des Schwellwertes, der das Bild in einen lesbaren Zustand versetzt.
X86\c_sharp\ECI X64\c_sharp\ECI	C#	Auswertung von ECI-Codes über einen QR Code
X86\c_sharp\Multi- Threading X64\c_sharp\Multi- Threading	C#	Sechs verschiedene Barcodetypen werden in einer definierbaren maximalen Anzahl gleichzeitiger Threads wiederholend gelesen.
X86\delphi\Linear	Delphi	Kommandozeilen-Programm: liest aus der Datei BarTest.tif 10 lineare Barcodes und listet die Ergebnisse
X86\delphi\Data Matrix	Delphi	Beispielprogramm für die Erkennung von Data Matrix Barcodes
X86\Java X64\Java	Java	Kommandozeilen-Programm: liest aus der Datei BarTest.tif 10 lineare Barcodes und listet die Ergebnisse. Der Quellcode enthält auch Abschnitte zur Lesung der 2D Barcodes DataMatrix und PDF417.
X86\VB	Visual Basic 6.0	Liest alle Dateien aus einem Quellverzeichnis der Reihe nach. Wenn ein Barcode gelesen wird, wird der Inhalt als Dateiname verwendet und die Datei in ein Zielverzeichnis bewegt.
X86\VB.NET X64\VB.NET	VB.NET	Dialog zur Eingabe eines Dateinamens, aus dieser Datei werden DataMatrix (oder, nach Entfernung von Programmkomentaren, PDF417 bzw. lineare) Barcodes gelesen. Die Ergebnisse werden in einer MessageBox angezeigt
X86\VB_MultiTiff	Visual Basic 6.0	Liest auf der angegebenen Datei lineare, DataMatrix und PDF417 Barcodes. Bei MultiTiff-Dateien wird in einer Schleife Seite für Seite abgearbeitet. Außerdem wird gezeigt, wie die Funktionen zur Ermittlung der aktuell gültigen Lizenz und zur Ermittlung der Version angesteuert werden. Der Prozess der Erkennung und die Barcode-Ergebnisse werden in einer Protokoll-Listbox ausgegeben.

Verzeichnis F_OCX\X86\sample (OCX - Schnittstelle)

Wie auch die F_DLL Schnittstelle wird die Variante QSBCOCX.ocx aus dem QS-Barcode SDK eingesetzt, wenn QS-Barcode SDK das Laden der Bilddatei eigenständig durchführen soll. In vielen Umgebungen ist ein OCX einfacher einzubinden als eine DLL und intuitiver zu nutzen. Die angegebene Image-Datei wird geöffnet und mit den übergebenen Barcodeparametern wird die Barcodesuche und Lesung durchgeführt.

Hinweise!	Dateiformate Die Beispielprogramme zur OCXFile-Schnittstelle unterstützen, sofern sie nicht mit einem festen Dateinamen arbeiten, viele Dateiformate. Eine komplette Liste finden in der QS-Barcode SDK Dokumentation im Anhang „14.1 Unterstützte Dateiformate“
	MultiPage-Dateien Bei mehrseitigen Dateien (MultiTiff oder Adobe PDF) wird normalerweise nur auf der ersten Seite nach Barcodes gesucht. Beachten Sie den Anhang „14.2 MultiPage Unterstützung“ aus der QS-Barcode SDK Dokumentation für Hinweise, wie Sie auch die weiteren Seiten durchsuchen können.
	Adobe PDF Dokumente Das Adobe PDF Format wird vom QS-Barcode SDK ab Version 3.8 unterstützt.



Beachten Sie den Anhang „14.3 Adobe PDF Dokumente – Spezielle Einstellungen“ aus der QS-Barcode SDK Dokumentation.

Unterverzeichnis	Sprache / Umgebung	Aufgabe
X86\vb6\linear\execute X86\vb6\linear\project	Visual Basic 6.0	Liest alle Dateien aus einem Quellverzeichnis der Reihe nach. Wenn ein Barcode gelesen wird, wird der Inhalt als Dateiname verwendet und die Datei in ein Zielverzeichnis bewegt. Die Pfade sind einstellbar, ebenso einige der Barcodeparameter.
X86\vb6\DynamicThreshold\execute X86\vb6\DynamicThreshold\project		Beispiel für die Verwendung des dynamischen Schwellwertes
X86\elo_script (nur de)	Elo Script Basic	Script, zeigt die Integrationsmöglichkeit von QS-Barcode in das Elo Office Archivsystem

Verzeichnis H_DLL\x86/x64\sample (Handle auf DIB - Schnittstelle)

Die Variante QSBarDLL_H.dll aus dem QS-Barcode SDK wird eingesetzt, wenn das Bild im Programm als DIB (Device Independent Bitmap) im Hauptspeicher vorliegt. Mit den übergebenen Barcodeparametern wird im Memorybereich die Barcodesuche und Lesung durchgeführt.

Unterverzeichnis	Sprache / Umgebung	Aufgabe
X86\c\Default X64\c\Default	C	Das Programm erwartet als Kommandozeilenparameter einen Dateinamen auf ein Bild im BMP-Format. Es wird ein Handle auf eine Windows Device Independent Bitmap (DIB) erzeugt. Diese DIB kann monochrom, mit Graustufen oder auch farbig sein. Diese DIB wird nach linearen Barcodes durchsucht und die gefundenen Barcodes auf der Standardausgabe ausgegeben. Die Barcodeparameter sind fest im Code eingestellt. Der Code zum Lesen von 2D Barcodes PDF417 bzw. DataMatrix ist in der Quelle (auskommentiert) vorhanden. Das Beispielprogramm wurde mit MS Visual Studio 2008 geschrieben und getestet.
X86\c\WhitePixel X64\c\WhitePixel	C	Die Bilddatei wird nach einem Interleaved 2/5 Barcode durchsucht. Der Funktion zum Suchen von Barcodes wird dabei eine spezifische Konfigurationsdatei (qsbci.ini) zur Bildverbesserung übergeben, die bestehende Fehler auf dem Bild vor der Suche korrigiert. Der gefundene Barcode wird auf der Standardausgabe ausgegeben. Die Barcodeparameter sind fest im Code eingestellt. Das Beispielprogramm wurde mit MS Visual Studio 2008 geschrieben und getestet.
X86\c\AdvancedSearch	C	Der Barcode kann im Standardfalle nicht gelesen werden. Vor der Lesung werden Reparaturen auf



X64\c\ AdvancedSearch		dem Barcode ausgeführt und der Barcode wird mit erweiterter Funktionalität gelesen.
X86\c\ DynamicThreshold X64\c\ DynamicThreshold	C	Die Bilddatei ist insgesamt zu dunkel für den Standard-Schwellwert (160). Daher erfolgt vor der Lesung eine dynamische Ermittlung des Schwellwertes, der das Bild in einen lesbaren Zustand versetzt.
X86\c_sharp\default X64\c_sharp\default	C#	Programm zur Eingabe eines Dateinamens. Aus dieser Datei werden lineare (oder, nach Entfernung von Programmkommentaren, PDF417 bzw. DataMatrix) Barcodes gelesen. Die Ergebnisse werden in einer MessageBox angezeigt
X86\c_sharp\Multi-Threading X64\c_sharp\Multi-Threading	C#	Sechs verschiedene Barcodetypen werden in einer definierbaren maximalen Anzahl gleichzeitiger Threads wiederholend gelesen.
X86\delphi	Delphi	Kommandozeilen-Programm: liest aus der Datei BarTest.bmp die linearen Barcodes und listet die Ergebnisse.

Verzeichnis \P_Lib\x86/x64\Sample (Pointer auf Memory - Schnittstelle)

Diese Variante aus dem QS-Barcode SDK wird eingesetzt, wenn das Bild im Programm als Bitmap im Hauptspeicher vorliegt. Mit den übergebenen Barcodeparametern wird im Memorybereich die Barcodesuche und Lesung durchgeführt.

Unter-verzeichnis	Sprache / Umgebung	Aufgabe
X86\c\Example1 X64\c\Example1	C	Das Programm erwartet als Kommandozeilenparameter den Dateinamen einer monochromen BMP-Datei (monochrom, weil hier die Schnittstelle p_lib zum Einsatz kommt. Beim Einsatz anderer Schnittstellen sind auch Graustufen- oder Farbbilder möglich). Diese Datei wird gelesen, nach Barcodes durchsucht und die gefundenen Barcodes auf der Standardausgabe ausgegeben. Die Barcodeparameter sind fest im Code eingestellt. Das Beispielprogramm wurde mit MS Visual Studio 2008 geschrieben und getestet.
X86\c\Example2 X64\c\Example2	C	Die Bilddatei nach wird nach einem Interleaved 2/5 Barcode durchsucht. Der Funktion zum Suchen von Barcodes wird dabei eine spezifische Konfigurationsdatei (qsbc.ini) zur Bildverbesserung übergeben, die bestehende Fehler auf dem Bild vor der Suche korrigiert. Der gefundene Barcode wird auf der Standardausgabe ausgegeben. Die Barcodeparameter sind fest im Code eingestellt. Das Beispielprogramm wurde mit MS Visual Studio 2008 geschrieben und getestet.
X86\c_dmatrix X64\c_dmatrix	C	Kommandozeilen-Programm: liest aus der Datei die 2D Barcodes vom Typ DataMatrix. Der Bilddateiname muss als Parameter übergeben werden.
X86\c_Multi-Threading X64\c_Multi-Threading	C	Sechs verschiedene Barcodetypen werden in einer definierbaren maximalen Anzahl gleichzeitiger Threads wiederholend gelesen.



X86\c_pdf417	C	Kommandozeilen-Programm: liest aus der Datei die 2D Barcodes vom Typ PDF417. Der Bilddateiname muss als Parameter übergeben werden.
X86\c_qrcode	C	Kommandozeilen-Programm: liest aus der Datei die 2D Barcodes vom Typ QR Code. Der Bilddateiname muss als Parameter übergeben werden.
X86\delphi	Delphi	Kommandozeilen-Programm: liest aus der Datei BarTest.bmp die linearen Barcodes und listet die Ergebnisse.

Viel Erfolg beim Integrieren. Bei Fragen hilft unser Support gern und schnell:

support@qualitysoft.de