NEWS · JUNE 18, 2026

# Kostenlose Barcode- und QR-Code-Generierung für Delphi VCL

By Dr. Holger Flick




Barcodes und QR-Codes sind überall. Produktetiketten, Lagerverwaltungssysteme, Rechnungen, Eintrittskarten, Asset-Tracking, mobile Deep Links — wenn Sie Business-Software entwickeln, wird früher oder später jemand danach fragen. Meine Freunde bei TMS Software — einem Unternehmen, für das ich sieben

Jahre lang als Evangelist arbeiten durfte — haben gerade eine fantastische neue Open-Source-Bibliothek veröffentlicht, die genau das für Delphi-VCL-Entwickler zu einem nativen Erlebnis ohne Abhängigkeiten macht.

Sie heißt **Bar&QRCodes4D** und ist vollständig kostenlos.

> **GitHub:** https://github.com/tmssoftware/Bar_QRCodes4D

## Warum natives Delphi wichtig ist

Bevor wir uns den Code ansehen, lohnt sich ein Blick darauf, was „natives Delphi" für Ihr Deployment tatsächlich bedeutet. Viele Barcode-Lösungen erfordern die Auslieferung einer externen DLL, den Aufruf eines Webservice oder eine Laufzeitumgebung von Drittanbietern. Bei Bar&QRCodes4D entfällt all das:

- Keine externe DLL, die ausgeliefert werden muss
- Keine serverseitige Generierung
- Keine Internetverbindung erforderlich
- Vollständiger Delphi-Quellcode enthalten
- Fügt sich direkt in bestehende VCL-Projekte ein
- Direkter PNG- und SVG-Export bereits integriert

Das ist ein makelloses Zeugnis für jede Unternehmensumgebung mit strengen Richtlinien für Abhängigkeiten.

## Erste Schritte mit Barcodes

Nachdem Sie die Units zu Ihrem Projekt hinzugefügt (oder das Package installiert) haben, ist das Rendern eines Barcodes so einfach wie das Ablegen einer `TTMSBarCode`-Komponente auf Ihrem Formular und das Setzen von zwei Eigenschaften:

```
uses
  TMSBarCode;

procedure TForm1.FormCreate(Sender: TObject);
begin
  TMSBarCode1.BarcodeType := tbcCODE128;
  TMSBarCode1.Value := 'TMS-123456789';
end;
```

Die Komponente rendert direkt auf dem VCL-Formular — keine Canvas-Tricksereien nötig.

## Unterstützte Barcode-Formate

Die Bibliothek deckt die gängigsten Barcode-Standards ab, denen Sie in Business-Anwendungen begegnen werden.

**EAN-13** — der internationale Standard im Einzelhandel:

```
procedure TForm1.GenerateEAN13;
begin
  TMSBarCode1.BarcodeType := tbcEAN13;
  TMSBarCode1.Value := '5412345678908';
end;
```

**Code 39** — beliebt in Industrie- und Logistikanwendungen:

```
procedure TForm1.GenerateCode39;
begin
  TMSBarCode1.BarcodeType := tbcCODE39;
  TMSBarCode1.Value := 'TMSCODE39';
end;
```

**Interleaved 2 of 5 (ITF)** — verbreitet in Versand und Distribution:

```
procedure TForm1.GenerateITF;
begin
  TMSBarCode1.BarcodeType := tbcITF;
  TMSBarCode1.Value := '1234567890';
end;
```

## Das Erscheinungsbild anpassen

Die Komponente bietet einen übersichtlichen Satz an Styling-Eigenschaften, mit denen Sie das Design Ihrer Anwendung oder die Vorgaben Ihrer Etikettenspezifikation treffen können:

```
procedure TForm1.SetupBarcodeStyle;
begin
  TMSBarCode1.BarColor        := clBlack;
  TMSBarCode1.BackgroundColor := clWhite;
  TMSBarCode1.ModuleWidth     := 2;
  TMSBarCode1.BarHeight       := 100;
  TMSBarCode1.Margin          := 10;
  TMSBarCode1.DisplayValue    := True;
end;
```

Ob Bildschirmdarstellung, Etikettendruck oder eingebettete Berichtserzeugung — diese Eigenschaften geben Ihnen genug Kontrolle, um die Ausgabe richtig hinzubekommen, ohne gegen die Komponente ankämpfen zu müssen.

## Validieren, bevor Sie drucken

Eines der Dinge, die ich als Entwickler am meisten schätze, ist elegante Fehlerbehandlung. Nicht alle Barcode-Formate akzeptieren beliebige Eingaben — EAN-13 verlangt genau 13 Ziffern, ITF erfordert numerische Zeichenketten gerader Länge, und so weiter. Bar&QRCodes4D stellt Ihnen eine `TryEncode`-Methode zur Verfügung, mit der Sie validieren können, bevor Sie einen Druckauftrag oder Export anstoßen:

```
procedure TForm1.ValidateBarcode;
var
  LPattern: string;
  LText: string;
begin
  if TMSBarCode1.TryEncode(LPattern, LText) then
    ShowMessage('Barcode erfolgreich generiert')
  else
    ShowMessage(TMSBarCode1.LastError);
end;
```

Dieses kleine Detail erspart eine Menge Frust, wenn der Barcode-Wert dynamisch aus Benutzereingaben stammt.

## Export nach PNG und SVG

Sie müssen einen Barcode in einen PDF-Bericht, einen E-Mail-Anhang oder eine Etikettenvorlage einbetten? Sowohl PNG- als auch SVG-Export sind direkt integriert:

```
// Raster-Export – ideal für E-Mail und direkten Druck
procedure TForm1.SaveBarcodeAsPNG;
begin
  TMSBarCode1.SaveToPNG('c:\temp\barcode.png');
end;

// Vektor-Export – ideal für skalierbaren Druck und Etikettendesign
procedure TForm1.SaveBarcodeAsSVG;
begin
  TMSBarCode1.SaveToSVG('c:\temp\barcode.svg');
end;
```

Der SVG-Export ist besonders wertvoll, wenn Ihr Workflow Etikettendesign-Tools oder hochauflösenden kommerziellen Druck umfasst — Vektorgrafiken skalieren verlustfrei auf jede Größe.

## QR-Codes sind genauso einfach

Die Komponente `TTMSQRCode` folgt derselben Design-Philosophie. Wert zuweisen — fertig:

```
uses
  TMSQRCode;

procedure TForm1.FormCreate(Sender: TObject);
begin
  TMSQRCode1.Value := 'https://www.tmssoftware.com';
end;
```

Richten Sie eine Handykamera darauf, und es funktioniert einfach.

**Die Generierung aus Benutzereingaben** ist ebenso unkompliziert — nützlich für URL-Shortener, Kontaktkarten, Produktabfragen oder jedes Szenario mit dynamischen Inhalten:

```
procedure TForm1.ButtonGenerateClick(Sender: TObject);
begin
  TMSQRCode1.Value := EditQRCodeText.Text;
end;
```

Und natürlich gilt dieselbe PNG/SVG-Export-API:

```
procedure TForm1.SaveQRCode;
begin
  TMSQRCode1.SaveToPNG('c:\temp\qrcode.png');
  TMSQRCode1.SaveToSVG('c:\temp\qrcode.svg');
end;
```

## Direkt aus der VCL drucken

Da die Komponenten nativ rendern, fügen sie sich mit `Vcl.Printers` direkt in die üblichen VCL-Druck-Workflows ein:

```
uses
  Vcl.Printers;

procedure TForm1.PrintBarcode;
begin
  Printer.BeginDoc;
  try
    TMSBarCode1.PaintTo(Printer.Canvas.Handle, 100, 100);
  finally
    Printer.EndDoc;
  end;
end;
```

Keine Wrapper-Bibliotheken, keine Zwischenformate — einfach natives VCL-Zeichnen auf den Drucker-Canvas. Einfacher geht es wirklich nicht.

## Installation mit TMS Smart Setup

Die Installation von Bar&QRCodes4D ist mit TMS Smart Setup ein Kinderspiel. Eine Zeile in der Eingabeaufforderung, und schon kann es losgehen:

```
tms install tms.barcode
```

## Fazit

Ich habe über Jahre beobachtet, wie TMS Software sein Komponenten-Ökosystem aufgebaut und verfeinert hat, und Bar&QRCodes4D ist ein großartiges Beispiel für das, was sie gut können: einen praktischen, alltäglichen Entwicklerbedarf erkennen und etwas Sauberes, von Abhängigkeiten Freies darum herum bauen. Wenn Sie VCL-Anwendungen schreiben, die mit Lagerbeständen, Logistik, Ticketing oder irgendeiner Form von Etikettendruck zu tun haben, verdient diese Bibliothek einen Platz in Ihrem Werkzeugkasten.

Sie ist Open Source, sie ist kostenlos, und sie ist pures Delphi. Greifen Sie zu.

> **Quellcode und Demo-Projekte:** https://github.com/tmssoftware/Bar_QRCodes4D

---

*Holger Flick ist unabhängiger Delphi-Berater, Dozent und GitKraken Ambassador bei FlixEngineering, LLC. Er war sieben Jahre lang Evangelist von TMS Software und arbeitet weiterhin intensiv mit dem TMS-Komponenten-Ökosystem.*

INDEPENDENT · AD-FREE · READER-SUPPORTED

# Free to read. Not free to make.

Every tutorial, article, and line of example code on Holger's Code is published without a paywall and without ads. This page is about what keeps it that way.

## Professional work, published free

The content here is held to the same standard as professional client work: every tutorial is built as a real project, tested against current versions, documented step by step, and revisited when the frameworks move on. A single in-depth tutorial routinely takes days — and that's before the bills behind it: servers, storage, build infrastructure, licenses, and the hardware everything is verified on.

If an article or tutorial here has saved you an afternoon of debugging, taught you a technique, or helped you ship, it created real value for you. Supporting it isn't charity — it's paying for professional work you found useful, at whatever price you choose. And if you can't, keep reading anyway. That's what the free part means.

## Ways to support

KO-FI

### Buy me a coffee

A one-time tip — no account, no subscription, any amount.

BOOKS & COURSES

### Buy the paid work

The paid tier of everything published free here — and you get something lasting in return.

SPREAD THE WORD

### Share an article

Send a tutorial to a colleague, cite an article, subscribe to the RSS feed.

Nothing here goes behind a paywall, whether you chip in or not. But if something on this site earned its keep today, this is where to say so.

**Support on Ko-fi —**