

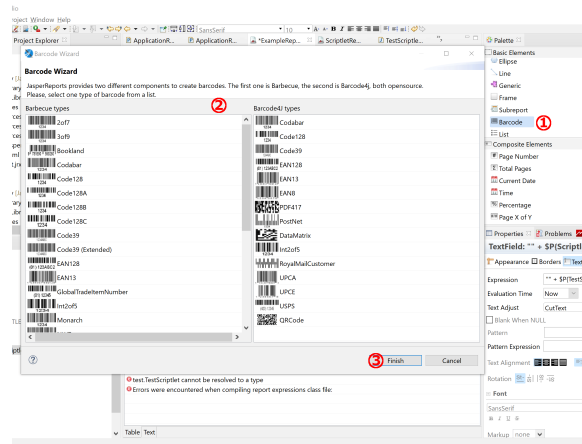
Barcode einbinden

- Barcode einbinden über das Image-Objekt
- Barcodes in Nuclos
- EAN13

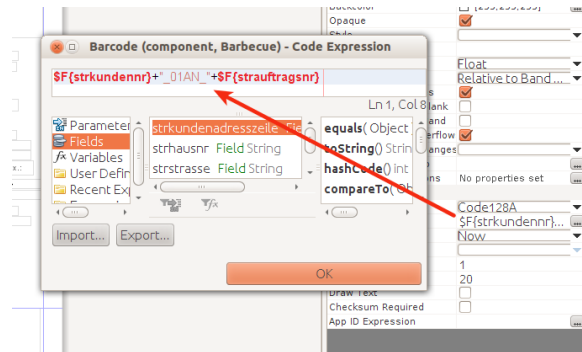
Es existieren mehrere Möglichkeiten Barcodes mit Jasperreports zu verwenden. In der Palette stehen unter der Komponente Barcode zwei Bibliotheken zur Verfügung [barcode4j](#) und [barbecue-1.5-beta1](#).

Einbinden eines Barcodes in einen Report

1. Barcodekomponente in der Palette auswählen und auf das Arbeitsblatt ziehen
2. Auswählen der gewünschten Barcode Bibliothek und eines Barcode Typen
3. Bestätigen per Klick auf *Finish*



In den Eigenschaften des Barcode-Feldes wird der Wert oder die Zeichenfolge, die der Barcode repräsentieren soll, eingegeben. Es gibt Barcodes die Zeichenketten in beliebiger Länge repräsentieren und andere die nur eine bestimmte Anzahl Zeichen akzeptieren.



Barcode einbinden über das Image-Objekt

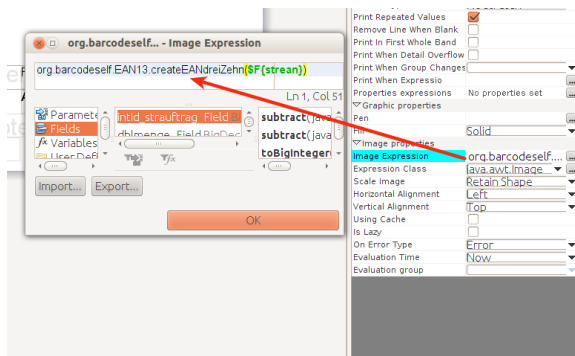
Die vorhergehende Methode ist die einfachste zum Einbinden eines Barcodes in ein Formular. Sollten dennoch Anforderungen entstehen, wie z.B. das Ändern der Schriftart und Größe des Textes unterhalb des Barcodes, kann der Barcode über das Image-Objekt eingebunden werden. In folgenden Beispiel wird davon ausgegangen, dass ein 13-stelliger Barcode übergeben wird, für welchen die Schriftart und Größe des Textes unterhalb geändert werden.

Dafür wird ein Java-Library Project mit einer Klasse in der folgenden Struktur angelegt:

Die [EANDreiZehn.jar](#), sowie die [barbecue-1.5-beta1](#) + [jdom.jar](#) werden dem Classpath der integrierten Jasperreports-6.17.0.jar, wie unter den nachfolgenden Überschriften beschrieben, zugänglich gemacht.

Im Formular-Designer wird das erwähnte Image-Element aus der Palette auf dem Arbeitsblatt platziert und die entsprechenden Eigenschaften wie im Bild links gesetzt.

- Image Expression = org.barcode4j.EAN13.createEANDreiZehn("12-stellig")
- Expression Class = java.awt.Image



Barcodes in Nuclos

In JasperSoft Studio können beide Barcode Varianten ohne Weiteres benutzt werden. Um sie in Nuclos zu benutzen, müssen die benötigten Abhängigkeiten hinzufügen. In folgendem werden die benötigten Abhängigkeiten für Barcode und Barcode4J aufgelistet.

Barcode

- [barcode-1.5-beta1](#)
- [jdom-1.0](#)

Barcode4J

- [barcode4j-2.1](#)
- [zxing-core-3.4.0](#)

Wie Sie diese Extensions genau zu Nuclos hinzufügen erfahren Sie in dem Kapitel [Jasper Extensions in Nuclos](#).

EAN13

Der EAN 13 (**E**uropean **A**rticle **N**umber) hat seinen Namen auf Grund seiner Länge von 13 Ziffern die er repräsentiert. Für die Verwendung im Formular ist zu beachten, das die letzte Ziffer die 13. eine Prüfziffer ist und bei der Generierung angefügt wird. Ergo es dürfen dem Barcodegenerator nur 12 Ziffern übergeben werden.

Sollte der EAN13 schon über einen externen Generator erstellt worden sein, ist die letzte Ziffer zu entfernen. Das kann bereits in der Datenquelle geschehen z.Bsp.: 'substring(ap."stream" from 1 for 12)" stream";' bei Postgres-Sql.

EANDreiZehn

```
/*
 * To change this template, choose Tools |
 * Templates
 * and open the template in the editor.
 */
package org.barcode4jEAN13;
import java.awt.Font;
import net.sourceforge.barbecue.Barcode;
import net.sourceforge.barbecue.BarcodeException;
import net.sourceforge.barbecue.BarcodeFactory;
import net.sourceforge.barbecue.BarcodeImageHandler;
import net.sourceforge.barbecue.output.
OutputException;
/**
 *
 * @author rufus
 */
public class EAN13 {
    public static java.awt.Image createEANdreizehn
(String c) throws OutputException, BarcodeException
{
    if(c.isEmpty() || c.equals("") || c.length()
!= 13){
        c = "0000000000000";
    }
    else{
        c = c.substring(0, 12);
    }
    Barcode b = BarcodeFactory.createEAN13(c);
    Font f = new Font("SanSerif", Font.BOLD,
12);
    b.setFont(f);
    return BarcodeImageHandler.getImage(b);
}
}
```

