Barcode für die ELO Automation Services

[Stand: 26.03.2019 | Programmversion: 12.00.000]

In den ELO Automation Services gibt es die Hilfsklasse *ELOAsUtils* mit Funktionen für das Auslesen und Schreiben von Barcodes. Für die Barcode-Funktionalität werden die Softek-Library und die ZXing-Library verwendet.

> **Information**: Weitere Informationen über die Softek-Library finden Sie unter:
>
> http://www.bardecode.com/en1/app/barcode-reader-toolkit-for-windows/
>
> Weitere Informationen über die ZXing-Library finden Sie unter:
>
> http://zxing.github.io/zxing/apidocs

Inhalt

1 Barcode auslesen mit der Softek-Library....2
1.1 Barcode-Formate der Softek-Library....2
1.2 Beispiel für das Auslesen eines QR-Codes....3
2 Barcode auslesen mit der ZXing-Library....4
2.1 Barcode-Formate der ZXing-Library....4
3 Barcode schreiben mit der ZXing-Library....6

1 Barcode auslesen mit der Softek-Library

Seit der Version 10 der ELOas ist das Auslesen von Barcodes über die Softek-Library möglich. Die Barcode-Erkennung erfolgt über die Softek Barcode-DLLs „SoftekBarcodeDLL.dll" und „SoftekBarcode64DLL.dll". Die Barcode-Erkennung wird in einem Skript folgendermaßen verwendet:

```
var barcodeReader =
Packages.de.elo.mover.utils.ELOAsUtils.createBarcodeReader2(emConnect);
var barcodeFile = new File("C://temp//BarcodeFile.tif");
var barcodeCount = barcodeReader.ScanBarCode(barcodeFile.getPath());
log.info("barcodeCount=" + barcodeCount);
var barcodeDescr = barcodeReader.GetBarString(barcodeCount);
log.info("barcodeDescr=" + barcodeDescr);
```

1.1 Barcode-Formate der Softek-Library

Die unterstützen Barcode-Formate finden Sie in der offiziellen Dokumentation der Softek-Library. Folgende Formate werden unterstützt:

- Codabar 1D
- Code 128 1D
- Code 2 of 5 Datalogic 1D
- Code 2 of 5 Iata1 1D
- Code 2 of 5 Iata2 1D
- Code 2 of 5 Industrial 1D
- Code 2 of 5 Interleaved 1D
- Code 2 of 5 Matrix 1D
- Code 3 of 9 1 D
- Code 3 of 9 Extended 1D
- Code 93 1D
- EAN-8 1D
- EAN-13 1D
- GS1-128, UCC-128, EAN-128 1D
- GS1-Databar 2D
- Patch Code Symbols 1D
- UPC-A 1D
- UPC-E 1D
- QR-Code 2D

- Data Matrix ECC200 2D
- Micro-PDF-417 2D
- PDF-417

1.2 Beispiel für das Auslesen eines QR-Codes

Das Auslesen von QR-Codes wird durch die folgende ELOas Anweisung aktiviert:

```
"barcodeReader.setReadQrCode(1);"
```



Information: Die Methoden der Klasse „BarcodeReader“ sind in der offiziellen ELOas Javadoc beschrieben.

2 Barcode auslesen mit der ZXing-Library

Das Auslesen von Barcodes erfolgt über die statische „getBarcode"-Methode. In dieser Methode werden die Datei, die Dateiseite mit dem Barcode und die Barcodekonfiguration als Parameter übergeben.

```
String barcode = ELOAsUtils.getBarcode(IXconnect ixConnect, File file, int
page, String barcodeConfig);
```

Die einzelnen Einstellungen in der Barcode-Konfiguration werden durch das Pipe-Symbol voneinander getrennt. Eine Beispielskonfiguration könnte wie nachfolgend beschrieben aussehen.

Beispiel

```
String barcodeConfig =
"POSSIBLE_FORMATS:CODE_128,QR_CODE|CHARACTER_SET:UTF8|ALLOWED_EAN_EXTENSION
S:2,5|PURE_BARCODE:TRUE|RETURN_CODABAR_START_END:TRUE|ASSUME_CODE_39_CHECK_
DIGIT:TRUE|TRY_HARDER:TRUE";
```

Zusätzlich gibt es in der Klasse *ELOAsUtils* die Methode „getBarcodeResult", die das komplette Barcodeergebnis zurückliefert.

Beispiel

```
Result barcodeResult = ELOAsUtils.getBarcodeResult(IXconnect ixConnect,
File file, int page, String barcodeConfig);
```

2.1 Barcode-Formate der ZXing-Library

Die unterstützten Barcode-Formate finden Sie in der offiziellen Dokumentation der ZXING-Library. Die einzelnen Formate sind in der Klasse *BarcodeFormat* aufgelistet. Weitere Informationen finden Sie auf:

http://zxing.github.io/zxing/apidocs/

Folgende Formate werden unterstützt:

- Aztec 2D
- CODABAR 1D
- Code 128 1D
- Code 39 1D
- Code 93 1D
- Data Matrix 2D
- EAN-13 1D

- EAN-8 1D
- ITF (Interleaved Two of Five) 1D
- MaxiCode 2D
- PDF417
- QR Code 2D
- RSS 14
- RSS EXPANDED
- UPC-A 1D
- UPC-E 1D
- UPC/EAN extension

3 Barcode schreiben mit der ZXing-Library

Das Erstellen von Barcodes in einer Dokumentenseite erfolgt über die Methode „writeBarcode“. In dieser Methode werden die Zieldatei, der Barcodetext, die Größe des Barcodes und die Barcode-Konfiguration als Parameter übergeben.

```
ELOasUtils.writeBarcode(IXconnect ixConnect, File targetFile, String
barcodeText, int width, int height, String barcodeConfig)
```

Beispielskonfiguration

```
String barcodeConfig =
"AZTEC_LAYERS:13|CHARACTER_SET:UTF8|DATA_MATRIX_SHAPE:FORCE_RECTANGLE|ERROR
_CORRECTION:M|MARGIN:20|PDF417_COMPACT:TRUE|PDF417_COMPACTION:NUMERIC|PDF41
7_DIMENSIONS:5,10,5,10";
```

Der Aufruf einer statischen AS-Methode in JavaScript:

```
var result = Packages.de.elo.mover.utils.ELOAsUtils.getBarcode(emConnect,
barcodeFile, 1, barcodeConfig);
```

Damit ist die Nutzung von Barcode Informationen auch in den ELO Automation Services möglich.