

Barcode.Swift (WASM)

C++ WASMエンジン Swift ラッパー

マニュアル

バージョン 1.0

有限会社 パオ・アット・オフィス

<https://www.pao.ac/>

目次

1. C++ バーコードエンジンを Swift から手軽に利用。
2. はじめに
3. 導入方法
4. クイックスタート
5. APIリファレンス
6. Pure Swift版との違い
7. 動作環境
8. ライセンス・お問い合わせ

C++ バーコードエンジンを Swift から手軽に利用。

ユーザーズマニュアル

バージョン 1.0 — 2026年2月

有限会社 パオ・アット・オフィス

<https://www.pao.ac/>

はじめに

C++ WASM版とは

Barcode.Swift (C++ WASMエンジン) は、C++ で書かれた高速バーコードエンジンを Node.js 経由 で Swift から利用するラッパーライブラリです。

C++ バーコードエンジンは Emscripten で WebAssembly にコンパイルされており、Node.js をサブプロセスとして起動して WASM を実行します。Swift 側は JSON で命令を送り、生成されたバーコード (Base64 PNG や SVG 文字列) を受け取ります。

```
Swift → JSON → node _barcode_runner.mjs → barcode.mjs → WASM → JSON → Swift
```

Pure Swift版と同じ18種のバーコードを生成でき、Foundation の Process と JSONSerialization のみで動作します。

```
import BarcodePaoWasm

let qr = QR(outputFormat: BarcodeFormat.png)
let result = try qr.draw(code: "https://www.pao.ac/", size: 300)
// result "data:image/png;base64,..."
```

特長

特長	説明
C++ 高速エンジン	実績あるC++バーコードエンジンをそのまま利用。高品質な出力
最小依存	Foundation のみ。Process で Node.js を起動、JSONSerialization でデータ交換
Node.js ブリッジ	Emscripten WASM を Node.js 経由で実行。インストールは node のみ
シンプルAPI	draw() が直接 Base64/SVG 文字列を返す。2ステップで完結

対応バーコード一覧

カテゴリ	バーコード
1D 工業用	Code39, Code93, Code128, NW-7 (Codabar)
1D 物流用	ITF, Matrix 2of5, NEC 2of5
GS1 系	GS1-128, GS1 DataBar 標準型/限定型/拡張型
商品コード	JAN-8 (EAN-8), JAN-13 (EAN-13), UPC-A, UPC-E
郵便	郵便カスタマバーコード
2D	QRコード, DataMatrix, PDF417

導入方法

動作要件

項目	要件
Swift	5.10 以上
Node.js	18 以上
OS	macOS 14+, Linux

ダウンロード

製品サイトからサンプルプロジェクトを含む ZIP ファイルをダウンロードしてください。

<https://www.pao.ac/barcode.swift/>

ファイル構成

```
barcode_pao_wasm/
├─ Package.swift
├─ Sources/BarcodePaoWasm/
│   ├─ Wrapper.swift      ← WASM
│   ├─ Code39.swift       ←
│   ├─ QR.swift
│   └─ ... (18)
└─ wasm/
    ├─ barcode.wasm        ← C++ WASM
    ├─ barcode.js          ← Emscripten
    ├─ barcode.mjs         ← ES Module
    └─ _barcode_runner.mjs ← Node.js
```

クイックスタート

QRコードをPNGで生成

```
import BarcodePaoWasm

let qr = QR(outputFormat: BarcodeFormat.png)
let result = try qr.draw(code: "https://www.pao.ac/", size: 300)
// result = "data:image/png;base64,..."
```

SVGベクター出力

```
let qr = QR(outputFormat: BarcodeFormat.svg)
let svg = try qr.draw(code: "Hello", size: 200)
// svg = "<svg xmlns=..."
```

REST APIサーバー (Vapor)

```
import Vapor
import BarcodePaoWasm

app.get("qr") { req -> Response in
    let code = req.query[String.self, at: "code"] ?? "Hello"
    let qr = QR(outputFormat: BarcodeFormat.png)
    let base64 = try qr.draw(code: code, size: 300)

    // Base64 Data URI JSON
    return Response(status: .ok, body: .init(string: "{\"base64\":\"\\(base64)\"}"))
}
```

APIリファレンス

共通メソッド

メソッド	説明
setOutputFormat(_ format: String)	出力形式を設定
setShowText(_ show: Bool)	テキスト表示 ON/OFF
setTextEvenSpacing(_ even: Bool)	テキスト均等配置
setForegroundColor(_ r: Int, _ g: Int, _ b: Int, _ a: Int)	前景色
setBackgroundColor(_ r: Int, _ g: Int, _ b: Int, _ a: Int)	背景色

出力フォーマット:

定数	値
BarcodeFormat.png	"png"
BarcodeFormat.jpeg	"jpeg"
BarcodeFormat.svg	"svg"

1次元バーコード共通メソッド

```
func draw(code: String, width: Int, height: Int) throws -> String
```

各バーコード型

クラス	描画メソッド
Code39, Code93, Code128, GS1_128, NW7, ITF, Matrix2of5, NEC2of5	draw(code:width:height:)
JAN8, JAN13, UPC_A, UPC_E	draw(code:width:height:)
GS1DataBar14, GS1DataBarLimited, GS1DataBarExpanded	draw(code:width:height:)
YubinCustomer	draw(code:height:)
QR, DataMatrix	draw(code:size:)
PDF417	draw(code:width:height:)

Pure Swift版との違い

項目	Pure Swift版	C++ WASM版
エンジン	Pure Swift	C++ (WASM)
依存	swift-png	Node.js
速度	速い	やや遅い（プロセス起動あり）
出力	Base64/SVG/Memory	Base64/SVG
導入	SPM のみ	SPM + Node.js
価格	¥22,000	¥11,000

動作環境

項目	要件
Swift	5.10 以上
Node.js	18 以上
OS	macOS 14+, Linux (Ubuntu 22.04+, CentOS Stream 9+)

ライセンス・お問い合わせ

使用許諾

本ソフトウェアは商用ライセンスです。ご購入いただいたライセンスに基づき、お客様の開発プロジェクトでご利用いただけます。

ライセンス

プラン	価格（税込）	内容
通常ライセンス	¥11,000	1開発者、無制限プロジェクト
3年サポート	¥9,900	メールサポート + アップデート
5年サポート	¥13,750	メールサポート + アップデート

お問い合わせ:

- Web: <https://www.pao.ac/>
- Email: info@pao.ac

Copyright (c) 2026 有限会社 パオ・アット・オフィス. All rights reserved.