

Introduction

PHP Library for ESC/POS Printers

Anonymous · 09/09/2026

PHP Library for ESC/POS Printers

last commit **august**  CI **passing** code size **58.5 KiB** open issues **0** downloads **348**
downloads **70 this month**

A PHP library for generating ESC/POS commands to communicate with thermal receipt printers.

Underneath, it uses [mike42/escpos-php](#) to talk to the printer, but that library is an implementation detail: nothing in this package's public API exposes a `Mike42\Escpos\*` type or constant. See [Architecture](#) for why that matters and how it is enforced.

Features

- Fluent, chainable API for common receipt printing tasks.
- Own vocabulary (enums and constants) for every ESC/POS value, instead of the underlying library's constants — see [Vocabulary](#).
- 1D and 2D barcode generation, including PDF417 (native command or as a printed bitmap when the printer has no native support) — see [Printing Images](#).
- Sending the generated bytes to a real network printer, without a raw socket or the underlying library's connectors — see [Network Delivery](#).
- Special character handling and replacement.
- Receipt cutting and cash drawer opening.

Installation

```
composer require derafu/escpos
```

Quick Start

```
use Derafu\Escpos\Enum\Justification;
use Derafu\Escpos\EscposPrinter;

// Initialize the printer. With no options, it buffers ESC/POS bytes in
// memory instead of talking to a real printer – useful for building the
```

```
// receipt and deciding what to do with the bytes afterwards.
$printer = new EscposPrinter();

// Print some text.
$printer
    ->println('Hello World!')
    ->setJustification(Justification::Center)
    ->println('Centered Text')
    ->println('Multiple lines')
    ->feed(2); // Add some empty lines.

// Finalize the receipt (adds final line breaks, cuts paper, opens the
// drawer if configured) and get the ESC/POS bytes.
$escposData = $printer->end();

// Do whatever you need with the bytes: return them from an API, save them
// to a file, or send them to a real printer (see Network Delivery).
echo $escposData;
```

Configuration Options

The `EscposPrinter` constructor accepts an options array as its first argument:

Option	Type	Default	Description
connector	array	<code>['type' => 'memory']</code>	Printer connector configuration. <code>memory</code> is the only type supported today — it buffers bytes so <code>dump()/end()</code> can return them.
profile	string	<code>'default'</code>	Printer capability profile (from Mike42's <code>capabilities.json</code>).
cut	bool	<code>true</code>	Automatically cut the paper when <code>end()</code> is called.
specialchars	bool	<code>true</code>	When <code>false</code> , accented characters (á, ñ, etc.) are replaced by their unaccented equivalent instead of being sent as-is.
pulse	array	<code>['pin' => 0, 'on_ms' => 120, 'off_ms' => 240]</code>	Cash drawer pulse, sent together with the cut when cut is true.

A second, optional constructor argument lets you replace the default backend — see [Architecture](#).

Full Example

The repository's `examples/full-example.php` exercises every feature (text formatting, image, barcode, PDF417, special characters). Run it with:

```
php examples/full-example.php | nc 172.16.1.5 9100
```

Note: you need netcat installed, and 172.16.1.5/9100 should be the IP and port of your thermal printer. See [Network Delivery](#) for a pure-PHP alternative to nc.

Last updated on 09/09/2026