

PForm2 — Report Engine Reference

A PHP report/print-form engine driven by plain HTML templates. Part of the Evol Soft toolset.

How it works

A template is an ordinary HTML page — a grid of cells, styled with normal CSS. One invisible column (column A, never printed) is used as a **marker column**: each row in that column carries a tag telling the engine what kind of row it is (header, repeating data row, totals, etc.). The engine walks the template top to bottom, and for every row marked as the repeating data block, it emits one output row per record in your data set — substituting `{ $field }` placeholders as it goes.

Blocks

A template is divided into named blocks, each a contiguous run of rows sharing one marker tag. Standard blocks include:

Block	Purpose
header	Printed once, at the top of the report.
colheads	Column headings for the data table, repeated on every page.
data	Repeated once per record in the data set.
subdata	A nested repeating block for master-detail reports (one or more detail rows per master row).
endReport / footer	Printed once, after all data rows — typically totals and signature lines.

Blocks are created and reordered visually in the Editor2 UI; the marker column is written automatically and never needs to be edited by hand.

Field syntax

A field placeholder is written as `{ $name }` , with optional colon-separated flags controlling formatting and behavior, e.g. `{ $total : SUM(1) , f2 }` .

Flag	Meaning
f2	Fixed to 2 decimal places (numeric formatting).
z	Suppress zero values (print blank instead of 0).
s	String/text mode.
nowrap	Disable text wrapping in the cell.

upper	Force uppercase.
date	Format the value as a date.

System fields like `{$_rownum__}` output the current row number within the repeating block.

Accumulators

Any numeric field can be marked to accumulate a running total by adding `SUM(N)`, where `N` is an accumulator level (levels let you keep several independent running totals — e.g. per group and grand total — in the same report). The accumulated value is then output anywhere in the template with `{SUM($field,N):f2}`. Multi-level accumulators combine with group breaks to produce subtotal-per-group plus a grand total, without any PHP code in the report itself.

Pagination

Page size is configured once (`setPageSize`); the engine paginates the data block automatically and repeats the `colheads` block at the top of every new page (`onStartPage`). A manual page break can also be forced from within a template when needed.

Master-detail & group breaks

Nested report sections (e.g. an order with its line items, or a product with the locations it's stored in) are handled with the `subdata` block and an `onSub` callback that supplies the detail rows for each master row. Group breaks (`checkBreak`) detect when a grouping field changes between rows and trigger a subtotal row automatically.

Editor2 workflow

1. **Add blocks** — lay out the named blocks (header, data, footer, etc.) that the report needs.
2. **Enter content** — type labels and `{field}` placeholders directly into cells.
3. **Format cells** — bold, borders, background/foreground color, number formatting.
4. **Adjust column widths** — drag column borders; widths are stored per block.
5. **Preview** — render the template against sample or live data before saving.
6. **Save** — writes the template back out as a plain HTML file with the marker column encoded.

Standard spreadsheet-style editing operations are supported throughout: insert/delete rows and columns, undo/redo, and keyboard navigation between cells (Tab for next cell, Enter for next row).

Rendering & output

A filled-in template renders to a paginated, printable HTML page by default. PDF export supports two engines: **TCPDF** (the default, a pure-PHP library, no external dependency) or a **headless-Chromium** engine for pixel-accurate rendering of more complex CSS layouts. A separate template parser handles plain ESC-code text output for dot-matrix and receipt printers, driven by the same underlying data.

