



Callback Implementation

Connect a saved report design to your database and produce finished PDFs.

REPORT

@field

A practical reference for the current editor and PHP class.

Contents

Every option below is accompanied by a practical example. Page numbers link to the corresponding section.

01 The separation of responsibilities	3
02 Install and call the class	3
03 Field syntax and callback arguments	4
04 A complete mysqli example	5
05 The two passes and page state	6
06 Formatting and page visibility	7
07 Images, QR and barcode data	8
08 Deployment and troubleshooting	8

01 The separation of responsibilities

A .ppr design describes the PDF page. PdfReport.php reads it and uses FPDF to render. Your callback returns data for tokens placed by the designer. The class does not open your database connection or invent a query.

Part	Responsibility	Example
.ppr design	Positions, fonts, page regions and field identifiers.	invoice.ppr contains @@customer and @@@description.
Your PHP code	Queries data, validates inputs and maps field identifiers to values.	mysqli_fetch_assoc() builds an array of item rows.
PdfReport class	Calls your function and writes the PDF with FPDF.	invoice.pdf is generated from invoice.ppr and your rows.

Layout changes

A designer can move, recolor or restyle existing fields without changing the PHP callback. When a new identifier appears in the design, add a matching branch to the callback.

02 Install and call the class

The class is in the ZioBitPhpPdfReport namespace. The attached source loads phpqrcode.php, fpdf/fpdf.php and barcode.php; include these with the class in a complete deployment. It also uses PHP extensions including mbstring, bz2 and iconv; images may require cURL and GD.

Constructor input	Meaning	Example
\$filename	Saved .ppr design path; required to render.	'invoice.ppr'
\$outputFilename	PDF filename; if provided, the constructor writes it to disk.	'invoice.pdf'
\$callback	Callable that receives every field request; a global callback() is the fallback if none is supplied.	'reportData'
\$nItems	Count of repeating rows to render; used for page count.	count(\$items)

Minimal call

```
<?php
require_once 'PdfReport.php';
use ZioBitPhpPdfReport\PdfReport;

function reportData($what, $index, $pdf, $sysVar) {
    if ($what === '__precalc_row__') return '';
    if ($what === 'customer') return 'Acme Ltd';
    if ($what === 'description') return 'Sample item';
    return '';
}

new PdfReport('invoice.ppr', 'invoice.pdf', 'reportData', 1);
```

Paths and output

Use paths that resolve from the PHP process working directory. The constructor sanitizes filenames, creates a local tmp directory if needed, and writes the PDF with FPDF Output("F", ...). It does not stream the PDF to the browser.

03 Field syntax and callback arguments

Identifiers use letters, digits and underscores after the prefix. The parser treats the three prefixes differently; use the field type that matches its location in the report.

Token	Purpose	Example and return
@@name	Fixed data requested from your callback.	@@customer -> 'Acme Ltd'
@@@name	Row data requested with a zero-based row index.	@@@description at index 2 -> 'Steel bolts'
##name	Built-in system value; no callback branch needed.	##currentPage -> 2

Argument	Value	Example
\$what	Identifier without @ signs, or __precalc_row__.	'description'
\$index	Zero-based row for @@@ fields. Fixed-field calls may use zero or a negative value; do not treat that as a row.	2 for the third row.
\$pdf	Current FPDF object.	Available if you need a supported FPDF operation.
\$sysVar	Array with currentLine, currentPage, totalPages, mode and linesPerPage.	\$sysVar['currentPage']

Placement matters

A repeating token is substituted while the renderer prints body rows. Use an Add DB Text object for @@@name.

Fixed @@name tokens can be placed in fixed text. The callback chooses by \$what; never derive a database row from a fixed-field index.

04 A complete mysqli example

This example assumes \$conn is an existing mysqli connection and an invoice_items table with description, qty and unit_price columns. Adapt the query and design fields to your schema. The design uses @@invoiceNo, @@customer, @@total, @@@description and @@@amount.

Load the rows, then instantiate the report

```
<?php
require_once 'PdfReport.php';
use ZioBitPhpPdfReport\PdfReport;

$invoice = [
    'number' => 'INV-2026-001',
    'customer' => 'Acme Ltd'
];
$items = [];
$invoiceId = 42;
$stmt = mysqli_prepare($conn,
    'SELECT description, qty, unit_price FROM invoice_items '
    . 'WHERE invoice_id = ? ORDER BY id');
mysqli_stmt_bind_param($stmt, 'i', $invoiceId);
mysqli_stmt_execute($stmt);
$result = mysqli_stmt_get_result($stmt);
while ($row = mysqli_fetch_assoc($result)) $items[] = $row;
mysqli_stmt_close($stmt);
$total = 0.0;
```

Implement every identifier in the design

```
function reportData($what, $index, $pdf, $sysVar) {
    global $invoice, $items, $total;
    if ($what === '__precalc_row__') {
        $r = $items[$index];
        $total += (float)$r['qty'] * (float)$r['unit_price'];
        return '';
    }
    if ($what === 'invoiceNo') return $invoice['number'];
    if ($what === 'customer') return $invoice['customer'];
    if ($what === 'total') return number_format($total, 2);
    if (!isset($items[$index])) return '';
    $r = $items[$index];
    if ($what === 'description') return $r['description'];
    if ($what === 'amount') {
        return number_format($r['qty'] * $r['unit_price'], 2);
    }
    return '';
}

new PdfReport('invoice.ppr', 'invoice.pdf',
    'reportData', count($items));
```

Generated template

Get PHP and PDF exports a PHP scaffold with branches for fixed text, repeating text, QR and barcode fields. Replace its TODO returns and the sample row count with your real values. The sample here uses a prepared mysqli statement and preloads rows before rendering.

05 The two passes and page state

The constructor first calls `__precalc_row__` once per item in calc mode. It then renders pages in render mode. This lets a first-page total use all rows without a second database query.

State key	Meaning	Example
mode	calc during pre-calculation, render while drawing.	Skip drawing side effects while mode is 'calc'.
currentLine	Zero-based index of the current data item.	4 for the fifth item.
currentPage	One-based page number.	2 on the second page.
totalPages	Total pages calculated from rows per page.	3 for a 53-row report at 23 rows/page.
linesPerPage	$\text{floor}((\text{page height} - \text{header} - \text{footer}) / \text{row height})$, at least 1.	A4: $\text{floor}((297 - 50 - 17) / 10) = 23$.

Keep pre-calculation deterministic. Do not add to a running total during ordinary field requests, because a field may be requested more than once. Return a string or scalar value for each field, and use empty string for a deliberately blank value.

System tokens: no callback required

Token	Result and example
##currentPage / ##page	Current one-based page number; Page ##page may become Page 2.
##last / ##total / ##totalPages	Last page count; Page ##page of ##totalPages may become Page 2 of 3.
##todayUSA / ##todayEU	Current date in m-d-Y / d-m-Y; e.g. 09-29-2026 / 29-09-2026.
##timeUSA / ##timeEU	Current time in h:i A / H:i; e.g. 11:15 AM / 11:15.

06 Formatting and page visibility

The design controls alignment, font, width, row height and page conditions. Callback values should be plain data; the renderer applies the saved style.

Option	Behavior	Example
Format	For repeating text, the renderer calls <code>sprintf(format, value)</code> before substitution.	<code>%.2f</code> turns 12.5 into 12.50.
Truncate Text	When Yes, cuts at Truncate At width and appends ... if needed.	Long product description in a 45 mm column.
Horizontal Align	Left, Center, Right or Justify within the saved text box.	Right-align an amount in a 28 mm box.
Print Where	Always, Never, First/Last, Odd/Even, or All Except First/Last.	Place a totals panel on Last Page.
Header / footer / row height	Determine repeating region and number of rows per page.	50 / 17 / 10 mm gives 23 rows on A4.

- Long values

A wide left-aligned value may wrap because the renderer uses MultiCell. Give row text enough width and height, or enable truncation, then inspect the generated PDF for overlap.
- Text encoding

The current renderer converts ordinary text to ISO-8859-1 for core FPDF fonts. THSarabunNew uses CP874; a field can also request CP874 explicitly. Test accented and non-Latin text in the actual PDF, as unsupported characters may be removed.

07 Images, QR and barcode data

The same token prefixes can supply QR and barcode content. For an Image URL object, the callback supplies a URL rather than the image bytes; the renderer downloads and caches the resolved URL during the request.

Object	Field and callback example	Important behavior
Fixed QR	Text @@paymentUrl; return 'https://pay.example/INV-1'.	Generated QR image links to its encoded content.
Row barcode	Text @@@sku; return \$items[\$index]['sku'].	Use an appropriate Barcode Type such as code-128.
Fixed URL image	URL @@logoUrl; return 'https://example.com/logo.png'.	Add this callback branch manually; URL image fields are not enumerated by the template generator.
Row URL image	URL @@@photoUrl; return \$items[\$index]['photo_url'].	Use the body row and an accessible image URL.
Embedded image	Image bytes are stored in the design.	Image Type and Quality affect conversion; no callback required.

Additional image URL branches

```
if ($what === 'logoUrl') {
    return 'https://example.com/assets/logo.png';
}
if ($what === 'photoUrl' && isset($items[$index])) {
    return $items[$index]['photo_url'];
}
```

Remote URLs

The URL renderer uses cURL and accepts the callback result as a network location. Return only trusted or validated addresses to avoid unexpected server-side requests. Test that your hosting can fetch each image and that required image-conversion functions are installed.

08 Deployment and troubleshooting

Deploy the complete editor and renderer package, not just the two attached source files. The editor also calls save_design.php, load_design.php, list_designs.php and generatephpsource.php for its save, load and export workflow.

Symptom	Likely cause	Check
File not found	.ppr path does not resolve from process cwd.	Use a correct relative path and verify the file exists.

Symptom	Likely cause	Check
Callback missing	No callable supplied and no global callback() exists.	Pass a callable as the third constructor argument.
Token visible in PDF	Identifier or object type does not match callback behavior.	Check @@ vs @@@, exact spelling and DB Text placement.
Blank image	URL is empty, inaccessible or not a valid image.	Check callback return, HTTP response and server cURL access.
Font or barcode error	Companion FPDF font or barcode/QR library is missing.	Install files required at the top of PdfReport.php.
Overlapping rows	Text wraps past the configured row height.	Increase Line data, widen the box or truncate text.
PDF not downloaded	The class writes a file; it does not emit a browser download.	Serve the produced file from your own PHP endpoint.

Current source behavior

The class renders portrait pages using saved width and height. It creates a shared local tmp folder and clears temporary image files at construction. Give concurrent jobs separate working directories or review this behavior for your deployment.

For every editor control and visual option, see the companion Graphic Editor manual. Both manuals document the attached editor 0.8.9 and PdfReport class as supplied.