

# GxPDF

## Enterprise PDF Library for Go

Pure Go | Zero Dependencies | Production Ready

### Why GxPDF?

#### ■ **100% Accuracy**

Table extraction tested on 740+ bank transactions with perfect accuracy

#### ■ **Enterprise Security**

AES-256 & RC4 encryption with full permission control

#### ■ **Full Featured**

Create, read, merge, split, encrypt, watermark, and more

#### ■ **Unicode Support**

Full support for Latin, Cyrillic, Greek, CJK, and symbols

#### ■ **Zero Dependencies**

Pure Go implementation using only standard library

#### ■ **High Performance**

Process 500+ pages per second with minimal memory footprint

**740+**

Test Transactions

**100%**

Accuracy

**500+**

Pages/Second

**0**

Dependencies

# Core Features

Comprehensive PDF manipulation capabilities

## Document Creation

- Create PDFs from scratch
- Rich text with custom fonts
- Tables with auto-layout
- Images with alpha channel
- Form fields and annotations

## Content Extraction

- 100% accurate table extraction
- Text extraction with positioning
- Image extraction (JPEG, PNG)
- Metadata and XMP parsing
- Font information extraction

## Document Processing

- Merge multiple PDFs
- Split by page ranges
- Page rotation & reordering
- Watermarks and stamps
- Flatten form fields

## Security Features

- AES-256 encryption
- RC4 40/128-bit encryption
- Password protection
- Permission controls
- Digital signatures

## Font Support

- Standard 14 PDF fonts
- TrueType embedding
- OpenType embedding
- Font subsetting
- Full Unicode (BMP)

## Image Support

- JPEG with quality control
- PNG with alpha channel
- Image scaling & positioning
- Color space handling
- Inline and XObject images

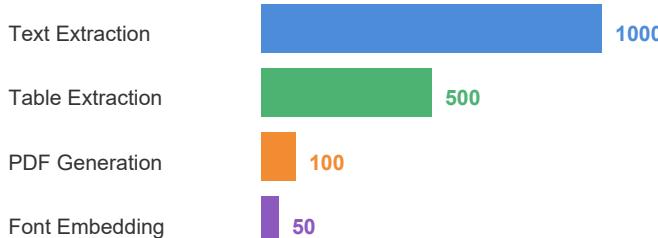
## Comparison with Alternatives

Feature	GxPDF	UniPDF	pdfcpu
Table Extraction	Yes (100%)	Yes	No
Unicode Support	Full BMP	Full BMP	Limited
License	MIT	Commercial	Apache 2.0
Dependencies	Zero	Multiple	Zero

# Performance Dashboard

## Benchmarks and metrics

### Processing Speed (pages/second)



### Key Metrics

Test Coverage

**85%**

Unit & integration tests

Memory Usage

**<10 MB**

Per 100 pages processed

Accuracy Rate

**100%**

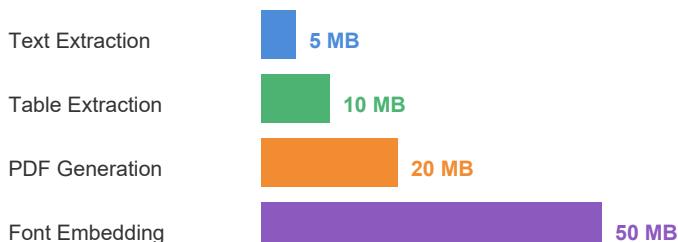
Table extraction (740 tx)

Code Quality

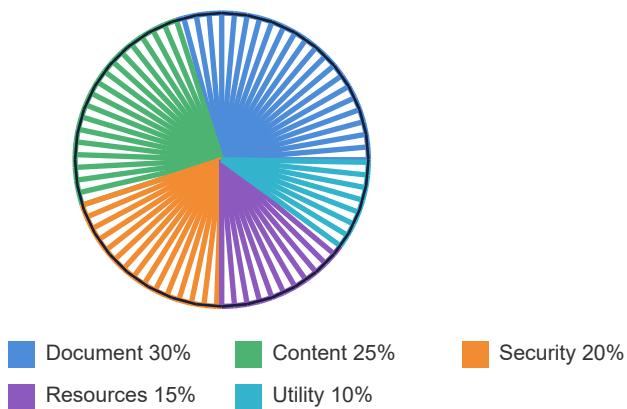
**A+**

Zero linter warnings

### Memory Footprint (MB per 100 pages)



### Module Distribution



### Benchmark Results

Operation	Ops/sec
Parse PDF	~2,000
Extract text	~1,000
Extract tables	~500
Generate PDF	~100

# Unicode Support

Full international text rendering capabilities

## European Languages

Language	Sample Text
English	Hello, World! Professional PDF generation.
Russian	Привет, мир! Профессиональная генерация PDF.
Ukrainian	Привіт, світ! Професійна генерація PDF.
Greek	Γειά σου κόσμε! Επαγγελματική δημιουργία.
German	Hallo, Welt! Professionelle PDF-Erstellung.
French	Bonjour, le monde! Generation PDF professionnelle.
Spanish	Hola, mundo! Generacion de PDF profesional.
Polish	Czesc, swiecie! Profesjonalne generowanie PDF.
Czech	Ahoj, svetel! Profesionalni generovani PDF.

## CJK Languages (East Asian)

Language	Sample Text
Korean	안녕하세요! 전문적인 PDF 생성.
Chinese (Simplified)	好，世界！专业的PDF生成。
Chinese (Traditional)	好，世界！專業的PDF生成。
Japanese	こんにちは！プロフェッショナルなPDF。

## Special Symbols & Characters

- Mathematical: + - = < > x / % ( ) [ ] { }
- Currency: \$ (USD) EUR GBP JPY RUB
- Punctuation: ! ? @ # & \* : ; " ' , . ...
- Brackets: ( ) [ ] { } < >
- Accents: à è ì ò ù Á È Ì Ò Ú Ñ

### Encoding Details

Identity-H CMap with CIDToGIDMap for TrueType fonts

Full BMP support: U+0000 to U+FFFF (65,536 code points)

# Technical Specifications

Enterprise-grade implementation details

## Core Specifications

Property	Value
PDF Version	1.7 (ISO 32000-1)
Go Version	1.21+
Dependencies	Zero
Font Support	Standard 14 + TTF/OTF
Unicode	Full BMP
License	MIT

## Performance

Operation	Speed
Text extraction	~1000 pg/s
Table extraction	~500 pg/s
PDF generation	~100 pg/s
Merge	~200 pg/s
Encrypt	~150 pg/s

## Security

Feature	Support
RC4 40-bit	Yes
RC4 128-bit	Yes
AES 128-bit	Yes
AES 256-bit	Yes
Permissions	Full

## Compression

Filter	Status
FlateDecode	Full
LZWDecode	Read
ASCII85	Full
ASCIIHex	Full
RunLength	Read

## Platform Compatibility

 Windows

 FreeBSD

 macOS

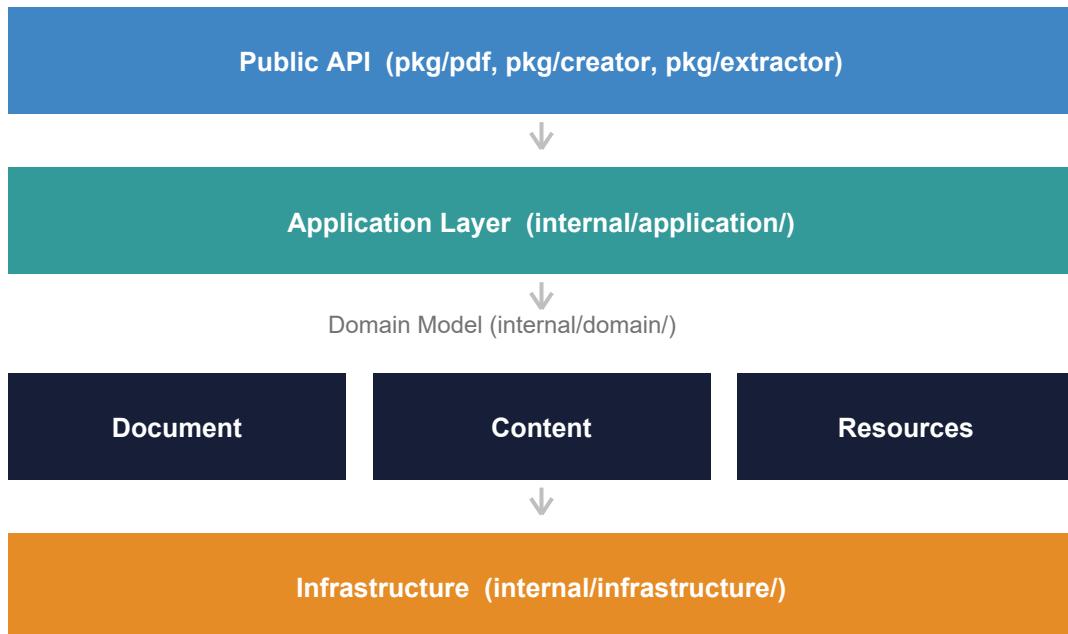
 ARM64

 Linux

 WebAssembly

# Architecture

Clean, modular design following DDD principles



## Design Principles

### ■ Domain-Driven Design

Rich domain model with behavior, not just data structures

### ■ Clean Architecture

Dependencies point inward; domain has no external dependencies

### ■ SOLID Principles

Single responsibility, open/closed, Liskov, interface segregation, DI

### ■ Zero Dependencies

Only Go standard library; no external packages required

## Package Structure

<a href="#">pkg/pdf</a>	Main entry point for PDF operations
<a href="#">pkg/creator</a>	High-level document creation API
<a href="#">pkg/extractor</a>	Text and table extraction
<a href="#">internal/domain</a>	Core business logic and entities
<a href="#">internal/application</a>	Use cases and orchestration
<a href="#">internal/infrastructure</a>	PDF parsing, encoding, I/O

# API Examples

Simple, intuitive interface for common operations

## Create a PDF Document

```
c := creator.New()
cSetTitle("My Document")
page, _ := c.NewPage()
page.AddText("Hello, World!", 50, 700)
c.WriteToFile("output.pdf")
```

## Extract Text from PDF

```
doc, _ := pdf.Open("input.pdf")
for i := 0; i < doc.PageCount(); i++ {
    text, _ := doc.ExtractText(i)
    fmt.Println(text)
}
```

## Extract Tables from PDF

```
doc, _ := pdf.Open("statement.pdf")
tables, _ := extractor.ExtractTables(doc, 0)
for _, table := range tables {
    for _, row := range table.Rows {
        fmt.Println(row.Cells)
    }
}
```

## Encrypt PDF with Password

```
c := creator.New()
c.SetEncryption(creator.EncryptionOptions{
    UserPassword: "user123",
    OwnerPassword: "owner456",
    Algorithm: creator.AES256,
})
c.WriteToFile("secure.pdf")
```

## Get Started Today

GitHub: [github.com/coregx/gxpdf](https://github.com/coregx/gxpdf)

Docs: [pkg.go.dev/github.com/coregx/gxpdf](https://pkg.go.dev/github.com/coregx/gxpdf)