Windows 11 Compilation Guide for Barkus

*PDF Barcode Splitter - Complete Setup for Fresh Environment*

# Overview

Barkus is a Python application that processes PDF documents to detect delivery number and customer name barcodes on each page, then splits the PDF into separate files based on these barcode combinations. This guide provides step-by-step instructions for compiling the application on a fresh Windows 11 environment with no existing Python installation.

# System Requirements

• Windows 11 (64-bit recommended)  
• Administrator privileges for installation  
• Internet connection for downloading dependencies  
• At least 2GB free disk space

# Step 1: Install Python 3.7+

## Download and Install Python

1. Visit **https://python.org/downloads/**  
2. Download Python 3.11 (recommended) for Windows  
3. Run the installer with these **IMPORTANT** options:  
 ✓ Add Python to PATH  
 ✓ Install pip  
 ✓ Install for all users (optional)  
4. Verify installation by opening Command Prompt and running:

python --version  
pip --version

# Step 2: Install System Dependencies

## Install Poppler (CRITICAL)

Poppler is **REQUIRED** for pdf2image to work:  
  
1. Download poppler-windows from:  
 **https://github.com/oschwartz10612/poppler-windows/releases/**  
2. Extract the ZIP file to **C:\poppler**  
3. Add **C:\poppler\Library\bin** to system PATH:  
 • Right-click "This PC" → Properties  
 • Advanced system settings  
 • Environment Variables  
 • Edit "Path" in System variables  
 • Add new entry: C:\poppler\Library\bin

## Install Visual C++ Build Tools

Required for compiling some Python packages:  
  
Option 1: **Microsoft C++ Build Tools**  
• Download from Microsoft Developer site  
• Install with C++ build tools workload  
  
Option 2: **Visual Studio Community**  
• Install with "Desktop development with C++" workload

# Step 3: Project Setup and Dependencies

## Create Project Environment

Navigate to your project directory and run:

# Navigate to project directory  
cd path\to\your\barkus\project  
  
# Create virtual environment  
python -m venv venv  
  
# Activate virtual environment  
venv\Scripts\activate  
  
# Upgrade pip  
pip install --upgrade pip  
  
# Install project dependencies  
pip install -r requirements.txt

## Key Dependencies Explained

|  |  |  |
| --- | --- | --- |
| **Package** | **Purpose** | **Special Requirements** |
| pikepdf | PDF manipulation and processing | None |
| pdf2image | Convert PDF pages to images | Requires Poppler |
| opencv-python | Computer vision for barcode detection | None |
| pyzbar | Barcode reading and decoding | May need libzbar DLL |
| numpy | Numerical operations | Version <2.0.0 required |

# Step 4: Compile to Executable

## Install PyInstaller

Install the PyInstaller package:

pip install pyinstaller

## Compilation Options

### Option 1: Single File Executable

Creates a single .exe file (slower startup):

pyinstaller --onefile --add-data "poppler;poppler" barkus.py

### Option 2: Directory Distribution (Recommended)

Creates a folder with executable and dependencies (faster startup):

pyinstaller --onedir --add-data "C:\poppler\Library\bin;poppler\bin" barkus.py

# Step 5: Testing and Distribution

## Test the Executable

1. Navigate to **dist\barkus\** directory  
2. Run **barkus.exe --help** to verify it works  
3. Test with a sample PDF file  
4. Verify barcode detection and PDF splitting functionality

## Distribution

The compiled application can now run on any Windows 11 machine **without requiring Python installation**. Simply copy the entire **dist\barkus\** folder to the target machine.

# Troubleshooting

## 'poppler' not found error

Ensure Poppler is installed and C:\poppler\Library\bin is in your system PATH. Restart Command Prompt after adding to PATH.

## Import errors during compilation

Make sure all dependencies are installed in your virtual environment. Run 'pip list' to verify.

## Missing DLL errors

Install Visual C++ Redistributable packages on the target machine if needed.

## Barcode detection not working

Verify that the input PDF contains valid barcodes and try adjusting the DPI parameter (--dpi flag).

# Usage Examples

Once compiled, use the executable with these commands:

# Basic usage  
barkus.exe input.pdf  
  
# Specify output directory  
barkus.exe input.pdf --output-dir C:\output  
  
# Handle pages without barcodes  
barkus.exe input.pdf --handle-no-barcode separate  
  
# Adjust DPI for better barcode detection  
barkus.exe input.pdf --dpi 600  
  
# Quiet mode  
barkus.exe input.pdf --quiet