Contents

1 REF Manager Documentation Tools

Building PDF Documentation from Markdown

Version: 2.0.0
Last Updated: November 3, 2025

1.1 Overview

This guide explains how to build PDF documentation from the markdown source files using the provided `build_docs.sh` script.

1.2 What Gets Built

The build script converts these markdown files to PDF:

1. `REF-MANAGER-README.md` - Main documentation
2. `QUICK-START-GUIDE.md` - Installation guide
3. `USER-GUIDE.md` - Feature guide
4. `TECHNICAL-DOCUMENTATION.md` - Developer reference
5. `TROUBLESHOOTING.md` - Problem solving
6. `CHANGELOG.md` - Version history
7. `DOCUMENTATION-INDEX.md` - Navigation guide
8. `README-TOOLS.md` - This file
9. `00-START-HERE.md` - Overview

Plus compiles LaTeX source files: - `QUICK-START-GUIDE.tex` - LaTeX version

1.3 Prerequisites

1.3.1 Required Software

1. **Pandoc** - Document converter

```
# Ubuntu/Debian
sudo apt-get install pandoc

# macOS
brew install pandoc

# Check installation
pandoc --version
```

2. **XeLaTeX** - LaTeX engine

```
# Ubuntu/Debian (full install recommended)
sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-
latex-extra

# macOS
brew install --cask mactex

# Check installation
xelatex --version
```

3. **PDF Tools** (optional, for combined PDF)

```
# Ubuntu/Debian
sudo apt-get install poppler-utils

# macOS
brew install poppler
```

1.4 Quick Start

1.4.1 Basic Usage

```
# Make script executable
chmod +x build_docs.sh

# Run the build script
./build_docs.sh
```

That's it! PDFs will be generated in `documentation/pdf/`

1.5 Build Process Details

1.5.1 What the Script Does

1. **Checks dependencies** - Verifies pandoc and xelatex are installed
2. **Creates directories** - Makes `documentation/pdf/` and `documentation/latex/`
3. **Converts markdown** - Uses pandoc to convert .md files to PDF
4. **Compiles LaTeX** - Compiles .tex sources with xelatex
5. **Combines PDFs** - Creates single combined PDF (if pdfunite available)
6. **Cleans up** - Removes auxiliary files

1.5.2 Build Options

Pandoc options used:

```
--pdf-engine=xelatex         # Use XeLaTeX for better Unicode support
--toc                         # Include table of contents
--toc-depth=3                 # Three levels in TOC
--number-sections             # Number all sections
-V geometry:margin=2.5cm      # Set margins
-V fontsize=11pt              # Set font size
-V linkcolor:blue             # Blue links
-V papersize:a4               # A4 paper
--highlight-style=tango       # Code syntax highlighting
-V mainfont="DejaVu Sans"     # Main font
-V monofont="DejaVu Sans Mono" # Monospace font
```

1.6 Output Structure

```
documentation/
├── pdf/                                          # Generated PDFs
│   ├── REF-MANAGER-README.pdf
│   ├── QUICK-START-GUIDE.pdf
│   ├── USER-GUIDE.pdf
│   ├── TECHNICAL-DOCUMENTATION.pdf
│   ├── TROUBLESHOOTING.pdf
│   ├── CHANGELOG.pdf
│   ├── DOCUMENTATION-INDEX.pdf
│   ├── README-TOOLS.pdf
│   ├── 00-START-HERE.pdf
│   └── REF-Manager-Complete-Documentation.pdf   # Combined (optional)
│
├── latex/                                       # LaTeX intermediate files
│   └── QUICK-START-GUIDE.tex
```

```
│
└── README.txt                                    # Info file
```

1.7 Customizing the Build

1.7.1 Editing the Script

The script can be customized by editing `build_docs.sh`:

Add/remove markdown files:

```
# Find this section in the script
MD_FILES=(
    "REF-MANAGER-README.md"
    "QUICK-START-GUIDE.md"
    # Add your files here
    "YOUR-NEW-FILE.md"
)
```

Add/remove LaTeX files:

```
# Find this section
TEX_FILES=(
    "QUICK-START-GUIDE.tex"
    # Add your .tex files here
)
```

Change PDF options:

```
# Modify the pandoc command
pandoc "$md_file" -o "documentation/pdf/$pdf_name" \
    --pdf-engine=xelatex \
    -V geometry:margin=3cm \    # Change margin
    -V fontsize=12pt \          # Change font size
    # Add more options...
```

1.7.2 Skip Certain Files

Comment out files you don't want to build:

```
MD_FILES=(
    "REF-MANAGER-README.md"
    # "TECHNICAL-DOCUMENTATION.md"  # Commented out
)
```

1.8 Troubleshooting Build Issues

1.8.1 Pandoc Not Found

Error:

```
✗ Error: pandoc not found
```

Solution:

```
# Install pandoc
sudo apt-get install pandoc

# Verify
pandoc --version
```

1.8.2 XeLaTeX Not Found

Error:

```
✗ Error: xelatex not found
```

Solution:

```
# Ubuntu/Debian - full install
sudo apt-get install texlive-full

# Or minimal install
sudo apt-get install texlive-xetex texlive-fonts-recommended

# Verify
xelatex --version
```

1.8.3 Font Errors

Error:

```
! Font \TU/DejaVuSans(0)/m/n/10=DejaVu Sans at 10.0pt not loadable
```

Solution:

```
# Install DejaVu fonts
sudo apt-get install fonts-dejavu

# Or change font in script
-V mainfont="Liberation Sans"
```

1.8.4 Build Fails on Specific File

Check: 1. File exists and is readable 2. File is valid markdown 3. No special characters causing issues 4. Check pandoc can read it: bash pandoc YOUR-FILE.md -o test.pdf

1.8.5 LaTeX Compilation Errors

Error:

```
! LaTeX Error: ...
```

Solution: 1. Check .tex file syntax 2. Ensure all packages are installed: bash sudo apt-get install texlive-latex-extra 3. Try compiling manually: bash xelatex QUICK-START-GUIDE.tex

1.8.6 Combined PDF Fails

Error:

```
pdfunite: command not found
```

Solution:

```
# Install poppler-utils
sudo apt-get install poppler-utils
```

1.9 Manual PDF Generation

1.9.1 Convert Single Markdown File

```
pandoc INPUT.md -o OUTPUT.pdf \
    --pdf-engine=xelatex \
    --toc \
    --number-sections \
    -V geometry:margin=2.5cm \
    -V fontsize=11pt
```

1.9.2 Compile Single LaTeX File

```
xelatex FILE.tex
xelatex FILE.tex  # Run twice for TOC
```

1.9.3 Combine Multiple PDFs

```
pdfunite file1.pdf file2.pdf file3.pdf output.pdf
```

1.10 Advanced Usage

1.10.1 Generate HTML Instead

```
# Modify script or run manually
pandoc INPUT.md -o OUTPUT.html \
    --toc \
    --standalone \
    --css=style.css
```

1.10.2 Generate DOCX

```
pandoc INPUT.md -o OUTPUT.docx \
    --toc \
    --reference-doc=template.docx
```

1.10.3 Generate EPUB

```
pandoc INPUT.md -o OUTPUT.epub \
    --toc \
    --epub-cover-image=cover.jpg
```

1.11 Build Script Reference

1.11.1 Command Line Options

The script takes no arguments currently, but you can modify it to accept options:

```
# Example modifications you could add:

# Clean only (no build)
./build_docs.sh --clean

# Build specific file
./build_docs.sh --file USER-GUIDE.md

# Verbose output
./build_docs.sh --verbose

# Custom output directory
./build_docs.sh --output /path/to/output
```

1.11.2 Exit Codes

- **0**: Success
- **1**: Dependency missing or build error

1.11.3 Environment Variables

You can set these before running:

```
# Use different LaTeX engine
export PDF_ENGINE=pdflatex
./build_docs.sh

# Change output directory
export DOC_OUTPUT_DIR=/custom/path
./build_docs.sh
```

1.12 Best Practices

1.12.1 Before Building

1. **Update all markdown files** with latest changes
2. **Check markdown syntax** - use a markdown linter
3. **Test manually** - convert one file first to test
4. **Check disk space** - PDFs can be large

1.12.2 After Building

1. **Verify PDFs** - open each to check formatting
2. **Check page numbers** - ensure TOC is correct
3. **Test links** - click internal links
4. **Review combined PDF** - check page order

1.12.3 For Distribution

1. **Compress PDFs** if needed:

   ```
   gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 \
      -dPDFSETTINGS=/ebook -dNOPAUSE -dQUIET -dBATCH \
      -sOutputFile=output-compressed.pdf input.pdf
   ```

2. **Create ZIP archive**:

   ```
   cd documentation
   zip -r REF-Manager-Documentation.zip pdf/
   ```

3. **Upload to documentation site** or share folder

1.13 Integration with Git

1.13.1 Pre-commit Hook

Create `.git/hooks/pre-commit`:

```
#!/bin/bash
# Auto-build docs on commit

echo "Building documentation..."
./build_docs.sh

# Add PDFs to commit
git add documentation/pdf/*.pdf

echo "Documentation built and staged"
```

1.13.2 GitHub Actions

Create `.github/workflows/build-docs.yml`:

```
name: Build Documentation

on:
  push:
    paths:
      - '**.md'
      - 'build_docs.sh'

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Install dependencies
        run: |
          sudo apt-get update
          sudo apt-get install -y pandoc texlive-xetex
      - name: Build documentation
        run: ./build_docs.sh
      - name: Upload PDFs
        uses: actions/upload-artifact@v2
        with:
          name: documentation
          path: documentation/pdf/
```

1.14 Scheduled Builds

1.14.1 Daily Build with Cron

```
# Edit crontab
crontab -e

# Add line for daily 2 AM build
0 2 * * * cd /path/to/ref-manager && ./build_docs.sh >> /path/to/build.log 2>&1
```

1.14.2 Weekly Summary Email

```
#!/bin/bash
# weekly-doc-build.sh

cd /path/to/ref-manager
./build_docs.sh

# Email results
mail -s "Weekly Documentation Build" admin@example.com < build.log
```

1.15 Tips and Tricks

1. **Preview before building**: Use a markdown preview tool to check formatting
2. **Keep builds fast**: Comment out files you're not changing
3. **Version PDFs**:

   ```
   cp documentation/pdf/USER-GUIDE.pdf \
      documentation/archive/USER-GUIDE-v2.0.pdf
   ```

4. **Watermark drafts**: Add draft watermark to PDFs in development
5. **Check file sizes**:

   ```
   ls -lh documentation/pdf/
   ```

   Large files may indicate embedded images that could be optimized

---

1.16 Support

Script issues? - Check dependencies are installed - Read error messages carefully - Try building one file manually - Check file permissions

PDF quality issues? - Adjust pandoc options - Try different font - Check source markdown - Review LaTeX output

Still stuck? - Check TROUBLESHOOTING.md - Contact system administrator - Submit issue on GitHub (if using)

1.17 Summary

To build all documentation:

```
./build_docs.sh
```

Output location:

```
documentation/pdf/
```

Requirements: - pandoc - xelatex (texlive) - poppler-utils (optional)

Build time: ~2-5 minutes depending on system

Version: 2.0.0
Last Updated: November 3, 2025
Script Author: George Tsoulas

For complete documentation, see the generated PDFs in `documentation/pdf/`.