# Markdown to Word Converter User Manual

## Table of Contents

1. Introduction

2. System Requirements

3. Installation

4. Configuration

5. Using the Converter

6. Supported Markdown Features

7. Word Styles

8. Troubleshooting

9. Best Practices

## 1. Introduction

The Markdown to Word Converter is a Python-based tool that converts Markdown documents into professionally formatted Microsoft Word documents. It maintains hierarchical styling and formatting while preserving the document's structure.

## 2. System Requirements

- Python 3.7 or higher

- Microsoft Word (2016 or later)

- Windows Operating System

- Required Python packages:

- pywin32

- pyyaml

- pytest (for testing)

## 3. Installation

### 3.1 Setting Up the Environment

```powershell

# Create and activate virtual environment

python -m venv venv

.\venv\Scripts\activate

# Install required packages

pip install pywin32 pyyaml pytest

```

### 3.2 Verifying Installation

```powershell

python tests/test\_environment.py

```

## 4. Configuration

### 4.1 File Structure

```

markdown\_to\_word/

├── src/

├── tests/

├── config/

└── output/

```

### 4.2 Word Template

The system requires a Word template (.dotm) with the following style hierarchy (0-9 levels):

- Heading 0-9

- Body 0-9

- Bullet 0-9

- Numbered 0-9

- Quote 0-9

- Code 0-9

## 5. Using the Converter

### 5.1 Basic Usage

```python

from src.converter import MarkdownToWordConverter

converter = MarkdownToWordConverter()

converter.convert(

template\_path="path/to/template.dotm",

markdown\_path="path/to/input.md",

output\_dir="path/to/output"

)

```

### 5.2 Output Files

- Output files are automatically named with timestamp: YYYYMMDD\_HHMMSS\_originalname.docx

- Files are saved in the specified output directory

## 6. Supported Markdown Features

### 6.1 Headers

```markdown

# Heading 1

## Heading 2

### Heading 3

```

### 6.2 Lists

```markdown

- Bullet point

\* Alternative bullet

+ Another bullet style

1. Numbered list

2. Second item

```

### 6.3 Text Formatting

```markdown

\*\*Bold text\*\*

\*Italic text\*

\*\*\*Bold and italic\*\*\*

`Code text`

```

### 6.4 Blockquotes

```markdown

> This is a blockquote

```

### 6.5 Code Blocks

````markdown

```

Code block content

Multiple lines supported

```

````

## 7. Word Styles

### 7.1 Style Hierarchy

- Each heading level (1-9) defines its own context

- All elements under a heading inherit its level

- Example:

```markdown

# Heading 1

Body text uses Body 1 style

- Bullet uses Bullet 1 style

## Heading 2

Body text uses Body 2 style

- Bullet uses Bullet 2 style

```

### 7.2 Style Fallbacks

1. Tries specific level style (e.g., "Body 3")

2. Falls back to base style (e.g., "Body 0")

3. Falls back to "Normal" if neither exists

## 8. Troubleshooting

### 8.1 Common Issues

1. \*\*Style Not Found Errors\*\*

- Verify template contains required styles

- Check style naming matches expected format

2. \*\*Import Errors\*\*

- Ensure virtual environment is activated

- Verify all required packages are installed

3. \*\*Word COM Errors\*\*

- Check Microsoft Word is installed

- Close any open Word documents

- Restart Word if necessary

### 8.2 Logging

- Log file: markdown\_to\_word\_debug.log

- Contains detailed operation information

- Use for troubleshooting conversion issues

## 9. Best Practices

### 9.1 Document Organization

- Use consistent heading levels

- Keep markdown formatting clean and standard

- Avoid mixing different list styles

### 9.2 Template Management

- Create separate templates for different document types

- Test templates with sample documents

- Keep style names consistent with expected format

### 9.3 Performance

- Close unnecessary Word documents

- Process one document at a time

- Regular saves during large document conversion

## Support

For additional support or to report issues:

1. Check the debug log file

2. Review error messages in the console

3. Verify style names in the Word template

4. Contact system administrator for template updates

---

\*\*Note:\*\* This manual assumes basic familiarity with Markdown syntax and Microsoft Word. For detailed Markdown syntax, please refer to the Markdown documentation.