

Apache OpenOffice Technical Editing

Editing Highlights

This document presents selected examples of technical editing completed for Apache OpenOffice user documentation. These examples demonstrate how editorial changes improved clarity, consistency, structure, usability, and reader comprehension across end-user documentation.

Editing types represented in this sample

Consistency editing

Content editing

Copyediting

Structural editing

Example 1: Consistency Editing

Before

Apache Open Office

After

Apache OpenOffice

Why this improved the content

This page 1 edit corrected the product name to match official project branding. Standardizing terminology across documentation improves consistency, professionalism, and reader confidence.

Example 2: Content Editing

Before

The fields you can use in Impress are:

After

The types of data you can use as fields in Impress are:

Why this improved the content

This revision made the description more precise. The original wording referred to fields generally, while the list actually described the types of data that can be used as fields.

Example 3: Copyediting

Before

However, the pasted text will probably not match the formatting of the surrounding text, or that of the other slides in the presentation. While this may be what you want on some occasions, in most cases you want to ensure the presentation style is consistent.

After

However, the pasted text may not match the formatting of the surrounding text or that of the other slides in the presentation. While this may be acceptable in some situations, in most cases, you want to ensure the presentation style is consistent.

Why this improved the content

This edit improved tone, precision, and readability. Replacing “probably” with “may” made the statement more accurate, while “acceptable” created a more professional tone.

Example 4: Structural Editing

Before

The section begins by telling users they do not need to be proficient in Basic, then presents the full macro example before explaining its key parts.

After

The revised section explains the role of Basic and key parts of the code before presenting the full macro example, with reassurance about user proficiency moved later.

Why this improved the content

This revision improved structure by giving readers context before the full code example and by moving reassurance to a more effective position. The result is a clearer and more instructional flow.

Conclusion

These examples reflect my approach to technical editing in a documentation context. My edits focused on improving clarity, consistency, usability, and organization while preserving technical meaning and supporting the needs of end users.

Project Repository: [My GitHub Repository](#)

Published Documentation: [[Apache OpenOffice PDF Wiki](#)]