Apache OpenOffice Technical Editing: Detailed Before-and-After Examples

This version provides deeper evidence for reviewers who want more than the one-page highlights.

Apache OpenOffice Technical Editing
Detailed Before-and-After Examples

This document presents selected examples of technical editing completed for Apache OpenOffice user documentation. The examples demonstrate how editorial revisions improved clarity, consistency, structure, readability, and overall usability across end-user documentation.

These samples represent several types of editing, including:

Consistency editing
Content editing
Copyediting
Structural editing
Each example includes the original wording, the revised version, and a brief explanation of how the edit improved the content.

Example 1: Consistency Editing

Before
Apache Open Office

After
Apache OpenOffice

Why this improved the content
This edit improved consistency by correcting the product name to match the official project branding. Standardizing product naming across documentation helps reduce confusion, supports professionalism, and ensures that users encounter the same terminology throughout the guides.

Although the change is small, this kind of consistency matters in large documentation sets. Repeated Example 1: Consistency Editing

Example 2: Content Editing

Before
The fields you can use in Impress are:

- Date (fixed).
- Date (variable) —updates automatically when you reload the file.
- Time (fixed).
- Time (variable)—updates automatically when you reload the file
- Author—First and last names listed in the Apache OpenOffice user data.
- Page number (slide number).
- File name.

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

- Date (fixed).

• Date (variable) —updates automatically when you reload the file.
• Time (fixed).
• Time (variable)—updates automatically when you reload the file
• Author—First and last names listed in the Apache OpenOffice user data.
• Page number (slide number).
• File name.

Why this improved the content

This edit improved content clarity by making the description more precise. The original wording referred generally to “fields,” but the list that followed actually described the types of data that can be inserted as fields in Impress.

The revision better aligns the introductory sentence with the information that follows. That improves user understanding by making the function of the list clearer and reducing conceptual ambiguity.

This kind of content edit is important because even small wording inaccuracies can affect how users interpret product features and instructions.

Example 3: Copyediting

Before

Text may be inserted into the text box by copying it from another document and pasting it into Impress. 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. There are several ways to ensure consistency; they're explained below.

After

Text may be inserted into the text box by copying it from another document and pasting it into Impress. 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. There are several ways to ensure consistency; they're explained below.

Why this improved the content

This edit improved sentence-level clarity, tone, and readability. Replacing “probably” with “may” makes the statement more precise and avoids overstating certainty. Changing “what you want” to “acceptable” creates a more professional tone that better suits technical documentation.

Minor punctuation and phrasing changes also improve flow by making the sentence read more smoothly. The revised passage preserves the original meaning while presenting the information in a more polished and reader-focused way.

This example demonstrates how copyediting can improve not just correctness, but also tone, professionalism, and overall readability.

Example 4: Structural Editing

Before

The section opens by reassuring users that they do not need to be proficient in Basic, then immediately presents the full macro example. Detailed explanation of the code appears afterward, requiring readers to encounter the full code sample before being guided through its key elements.

After

The revised section introduces the role of Basic more directly, explains the most important parts of the code before presenting the full macro example, and moves reassurance about user proficiency later in the section. This gives readers context before they encounter the full example and creates a more guided instructional flow.

Why this improved the content

This edit improved the structure by reorganizing the section so that the explanation comes before the full macro example rather than after it. In the original version, readers were asked to process the code before receiving enough guidance about what to notice or why it mattered. In the revised version, the section first introduces the topic, explains key concepts such as rem, dim, and the document variable, and only then presents the full macro.

The revision also moves the reassurance that proficiency in Basic is not required to a later point in the section, where it supports the explanation without weakening the opening. As a result, the section becomes easier to follow, more instructional in tone, and better organized around reader comprehension.

Although the change is small, this kind of consistency matters in large documentation sets. Repeated Example 1: Consistency Editing

Conclusion

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

Together, these examples show how technical editing strengthens documentation by making content easier to understand, more consistent to navigate, and more effective for the intended audience.

Project Repository: My GitHub Repository

Published Documentation: [Apache OpenOffice PDF Wiki]