CiviCRM documentation style guide

This document defines the standards for CiviCRM documentation, including grammar, formatting, and more. Our documentation should prioritize clarity, consistency, and accessibility while reflecting the open-source, community-driven nature of the project.

Structure

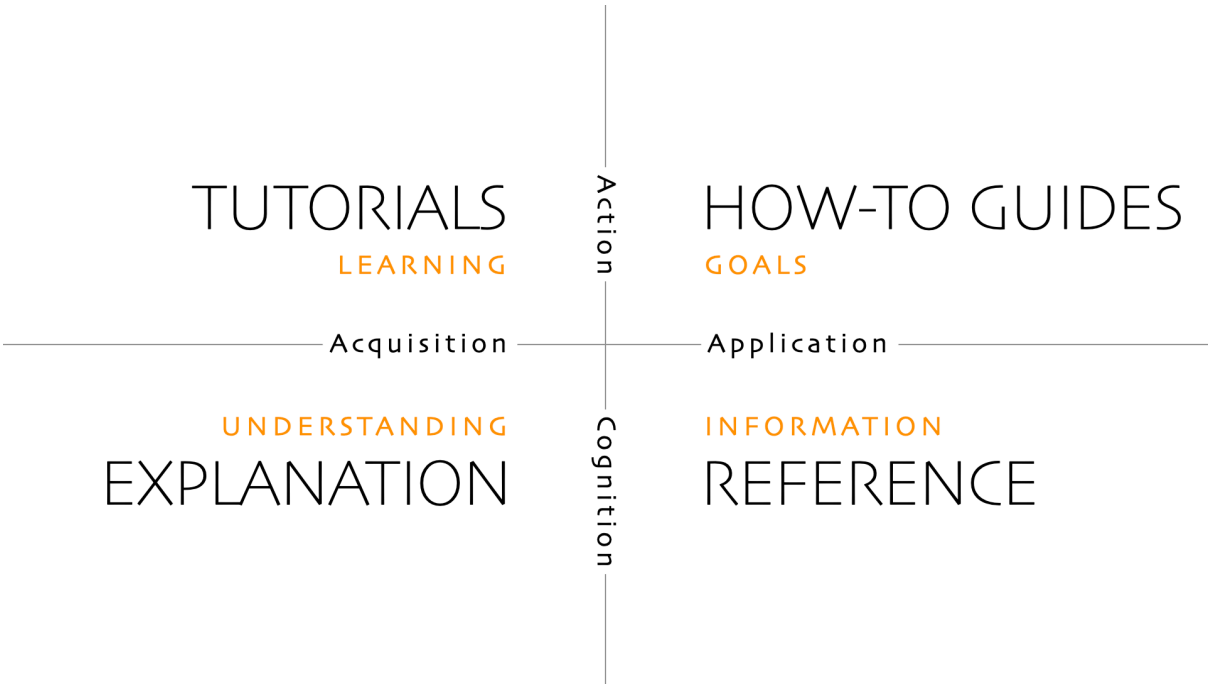
We have adopted the Diátaxis Framework. The core idea of Diátaxis is that there are fundamentally four identifiable kinds of documentation that respond to four different needs. The four kinds are: *tutorials*, *how-to guides*, *reference* and *explanation*. Each has a different purpose, and needs to be written in a different way.

A tutorial is a lesson that takes a student by the hand through a learning experience.

A how-to guide addresses a real-world goal or problem, by providing practical directions to help the user who is in that situation.

Reference guides contain the technical description - facts - that a user needs in order to do things correctly: accurate, complete, reliable information, free of distraction and interpretation.

Explanatory guides provide context and background. They serve the need to understand and put things in a bigger picture.



Tone and voice

The goal is to provide information that's easy to search and scan for all users within our community as well as people that are evaluating the product. We should aim to be:

- Clear, Concise, and Friendly: Aim for a professional but approachable tone.
- Community-Oriented: Use inclusive language that welcomes users of all skill levels.
- Encouraging: Guide the user—don't talk down to them.

Don't try to write exactly the way you speak; you probably speak more colloquially and verbosely than you should write, at least for developer documentation. But aim for a conversational tone rather than a formal one.

- Use active voice: "Click the Save button" instead of "The Save button should be clicked."
- Avoid jargon: Use plain language and explain technical terms when first introduced.
- Be task-focused: Documentation should help users do something, not just explain how it works.

Consider that readers come from many different cultures and may have varying levels of ability reading English. As much as possible, avoid culturally specific references. Simple and consistent writing can also make it easier to translate documents into other languages.

Accessibility

The World Health Organization estimates that 15% of the world's population (more than 1 billion people) have an accessibility need. When documentation is written with accessibility in mind, it improves the overall experience for all readers.

General dos and don'ts

- Don't use ableist language. Avoid bias and harm when discussing disability and accessibility.
- Avoid unnecessary font formatting. (Screen readers explicitly describe text modifications.)
- Don't force line breaks within sentences and paragraphs. Line breaks might not work well in resized windows or with enlarged text.
- Avoid when possible camel case and all caps. Some screen readers read capitalized letters individually, and some languages are unicase. Follow capitalization guidelines.
- Depending on the screen reader (or personal settings), not all punctuation marks are read. Make sure the same meaning is conveyed to the reader without punctuation marks. So avoid if possible the use of exclamation marks, question marks, and semicolons.
- Don't use *&* instead of *and* in headings, text, navigation, or tables of contents. However, it's OK to use *&* when referencing UI elements that use *&*, or in table headings and diagram labels where space constraints require abbreviation. Of course, it's fine to use & for technical purposes in code.

Formatting rules

- Keep titles concise.
- Lists: Use bullet points or numbered steps when appropriate.
- Inline Code: Use backticks for commands, paths, or variables (e.g., civicrm_api3).
- Code Blocks: Use fenced code blocks (triple backticks) for code examples.

Capitalisation

Don't use unnecessary capitalization; before you capitalize a word, think about why (and whether) it should be capitalized. In document titles and headings, use sentence case. That is, capitalize only the first word in the title, the first word in a subheading after a colon, and any proper nouns or other terms that are always capitalized a certain way.

Capitalize product names.

Use sentence case for captions, labels, and other text in images and diagrams.

Use lowercase for glossary and index terms unless the term is a proper noun or has another reason to require capitalization.

Words, names and terms

Use CiviCRM (not “Civi CRM” or “civiCRM”).

Refer to Contacts, Activities, Contributions, etc., using their proper noun form.

Use Administrator for backend users and User for frontend users when roles differ.

Use neutral examples that apply across regions.

Other common terms

In general we will use standard American English spelling. We would like to standardise on the following spellings for common technology terms:

- email
- online
- setup (noun), set up (verb)
- backup (noun), back up (verb)
- login (noun), log in (verb)
- web
- website
- internet
- virtualization
- space-separated, comma-delimited

Break the rules

> *Break any of these rules sooner than say anything outright barbarous.*
>
> —George Orwell, "Politics and the English Language"

This guide contains guidelines, not rules. Depart from it when doing so improves your content.

For example, if we recommend spelling a term as one word, and you determine that the hyphenated version of a term in your domain is more appropriate for your readers, then it's fine to use that instead. We acknowledge that sometimes there are competing forms of the same word in wide use, especially as new terms emerge, and you might have good reasons for departing from this guidance.

When you depart from this guide, be consistent throughout your document.

========== Other Ideas below here ==========

Use of AI

Use AI to help produce documentation and refer it to this guide with additional prompts as appropriate. This way you can produce brief and reasonably rough documentation and turn it into fleshed out pages which you can tidy up. This process can save hours of writing time. If the output isn't quite right, try reframing the request with amended prompts. You do need a human in this process!