

# The KDE Style Guide

## The KDE documentation team

Revision 0.01.00 (2002-05-18)

Copyright © 2002 The KDE e.V.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

A quick reference guide to writing documentation, KDE style. standards

---

### Table of Contents

#### 1. Introduction

#### 2. Consistency

Dates and Revision Numbers

Other Consistency Issues

#### 3. Writing Well

#### 4. Other general tips and hints for good writing

#### 5. Resources

General Purpose Websites

Dictionary Sites

Thesaurus Sites

Style Guides and Other Resources

#### 6. Credits and Licenses

## Chapter 1. Introduction

The purpose of writing documentation for the KDE Project, is to create for all the users, a complete, well organized set of documentation for each and every component of the KDE project. In order to achieve this goal, we have drafted the following guide to help.

This document is very new, and at the moment, very sparse.

If you have comments or additions, please do not hesitate to suggest them on the [kde-doc-english@kde.org](mailto:kde-doc-english@kde.org) mailing list.

In the meantime, here are some short notes based on content that was previously on the [KDE i18n](#) website.

## Chapter 2. Consistency

### Table of Contents

Dates and Revision Numbers

Other Consistency Issues

## Dates and Revision Numbers

While it may seem like a minor issue, and a minor part of your document, it is very important that the following guidelines are adhered to:

- All dates which are part of the text of your document should be spelled out i.e. ?March 2, 2000?

This is the only way to be sure that ?03/02/2000? is interpreted correctly in all languages, and by all readers.

### Important

Exception: in the `date` tag, dates should be in the ?YYYY-MM-DD? format.

- All information included between the `<releaseinfo>` and `</releaseinfo>` should match the release number of the application.

## Other Consistency Issues

- Please submit only plain ASCII text, or Docbook XML?. Anything else will be returned to you. (Docbook XML? is preferred.)
- Make sure you first read, and follow, the documentation template provided. You can find this in the source folder for KDE. Simply point a browser to `kdelibs/doc/kdoctools/template.docbook`

### Note

If there is existing documentation (from the developer, or from KDE 1.x or 2.x), then use that as a base to work from, but it needs to conform to the layout from the documentation template.

- Spell things according to Standard American spelling, except for proper names, places, etc.
- Make sure to set your spellchecker to US English. Make sure you *use* your spellchecker.
- If there is a non-English word, which is used in an English sentence, be sure to spell this correctly, using appropriate accent marks, and any special characters. Use the KCharSelect application if you don't have the correct keys on your keyboard.

- Abbreviations should be capitalized, unless they are specifically intended to not be capitalized. (i.e. is a good example).
- Punctuation within numbers should be Americanized: 10,000.00
- It is more legible to use written numbers where the number has no technical value, e.g. ?There are three buttons on the dialog?. In this context ?3? is not technically significant. Numbers with technical significance should be written as figures. (i.e. ?a 10 MB file?, not ?a ten MB file?.)
- Spellcheck and proofread your work before you send it in. Or even better, have someone else read it over. Spellchecking programs won't catch incorrect words such as using ?their? for ?there?.

## Chapter 3. Writing Well

Some things to think about when writing documentation

How many times have you attempted to read a man page, and given up in frustration ? all the facts are there, but the writing is too dry and technical for you to make sense of them? We want to avoid this with the documentation you're writing.

It's difficult to avoid the very dry, choppy style we're all too familiar with, but do try to make the writing flow and keep it *user-friendly*. Try to be as friendly as you can manage without being patronizing, and without sacrificing clarity or detail.

Things you should consider:

- Make sure you explain all aspects of the interface, and that you don't leave out any command line options available.
- It's important to keep a logical progression of difficulty. Keep the first few pages of the document simple, and accessible to users who have never seen the application before. More technical information should appear towards the end of the document.
- Be sure to write to the level of the intended reader. By this we mean, a Samba control module has a very different user base than a user of a game, and the manuals should reflect this. Don't insult the Samba networker's intelligence, and don't assume knowledge for the gamer that might not be there.
- It is highly encouraged to use screenshots, pictures of action buttons, etc. These are GUI applications, and a picture can be worth a thousand words.
- Please define computer abbreviations, unless they are well understood by all computer users (There's no need to define RAM, PC, etc.).

For example, ?If you are going to use PPP (Point to Point Protocol), then.....?. Define it only once though, in this example you could now use PPP without explanation for the rest of the document.

The first time you introduce the word you should use `<firstterm>` or consider setting up a glossary and using `<glossentry>`.

- Remember the small things: If it's dockable, explain how to do that. Don't forget to mention where it installs itself in the K menu. If there are any environment settings required, and if they must be set manually, explain how to do that - if they're set automatically, they still need to be documented.
- Write something you would be happy to read as a user who doesn't know how the application works. Perhaps if you have a friend or family member who isn't familiar with the application you're writing about, have them work through using it, with your documentation as a guide.

## Chapter 4. Other general tips and hints for good writing

A good way to catch simple errors is to read the text out loud, or have someone else read it to you. Passages that don't "flow" or have obviously awkward construction of the type you may miss on the screen, will usually become blindingly obvious when you hear them. This is especially the case with detecting really long sentences, as you will run out of breath and turn blue.

Be concise, be specific, and be direct. Choose your words carefully, and be certain you know the exact meaning of every word you use. Is it the right word? Use a dictionary, and find out.

Use complete sentences. Not fragments. Like these ones.

While talking about sentences:

- Avoid run on sentences, sentences that cover several different subjects, or sentences that could be broken up into several sentences; avoid sentences that can fill a whole paragraph all by themselves and that are really long, like this one, which is all of the above.
- Use a comma before "and" in compound sentences, e.g. "Use the left mouse button to select and copy text, and the middle mouse button to paste it."
- Keep to logical sentence order.

For example, "Konqueror is a web browser with the ability to browse file systems and it includes a javascript interpreter." (Do you see why this is awkward?)

- Try not to use the same word several times in the same sentence. An exception to this, is an application command or technical word, where this repetition is necessary, and improves clarity.
- Do not start sentences with any of "and," "so," "but," "because," or "however."
- Try to avoid contractions, rather spell out both words; e.g., "it is" rather than "it's"; "can not" rather than "can't"
- There is no need to worry about simple text formatting such as leaving two spaces after punctuation or indenting paragraphs. This is all handled by Docbook XML and the XSLT stylesheets in use.

Remember, we have also an active proofreading team, and there is always someone to help you with grammar, so just write and have fun!

## Chapter 5. Resources

### Table of Contents

#### General Purpose Websites

Dictionary Sites

Thesaurus Sites

Style Guides and Other Resources

## General Purpose Websites

A few sites you may find worth bookmarking, or at least visiting.

### Dictionary Sites

- Merriam Webster Online
- Dictionary.com

### Thesaurus Sites

- Merriam-Webster have a thesaurus as well as a dictionary
- Roget's Thesaurus

### Style Guides and Other Resources

- Strunk & White is the base style guide for many newspapers and press galleries. If you want to look up a grammar or usage question, this is the place to go.
- The alt.usage.english newsgroup. If you'd like every sentence chewed over, dissected, and then rewritten several conflicting ways, or some great advice about bacon, you'll find both here. Many real live editors and authors hang out here, and they do know their stuff. Just make sure you read all ten or so FAQ's before asking a question.

Other possibly useful sites

- The Wired Style Guide Bookmarks section
- [http://www2.rsc.ccc.tn.us/~jordan\\_jj/OWL/Clutter.html](http://www2.rsc.ccc.tn.us/~jordan_jj/OWL/Clutter.html) If you tend to write very wordy sentences, here's a page with some help for you. Includes a useful table of ways to rewrite common mistakes.

- <http://owl.english.purdue.edu/Files/116.html> is a page about how to make sentences ?flow? better

## **Chapter 6. Credits and Licenses**

The KDE Documentation Style Guide

Copyright 2002, Lauri Watts, Mike McBride, Eric Bischoff, Frederik Fouvry.

This documentation is licensed under the terms of the [GNU Free Documentation License](#).