

---

# OpenOffice.org User Interface Text Style Guide

Dec 2005, [Version](#) 2.19

Author: Elizabeth Matthis; Modified by Christian Jansen

---

## 1. About This Guide

---

### Purpose

This guide describes how to design text for the user interface (UI) of the English version of OpenOffice.org. There are also notes when special guidelines are needed for German, because OpenOffice.org is designed and implemented in German and English simultaneously. Any guidelines relating to German are highlighted by the colors of the German flag in the right margin, as seen to the right of this paragraph.



For the purposes of this guide, the definition of text on the user interface only includes menus and submenus, dialogs, tooltips (short ones displayed during “What's This?” mouse-over), message boxes, wizards, etc., but **not** Command Line Interface, Help or documentation, even though they also appear onscreen and thus constitutes, in some definitions, part of the user interface.

### Audience

This guide is predominantly for the people who design the German and English UI text of OpenOffice.org: mainly user experience designers and user interface linguists. Additional people for whom these guidelines will be important are those working in quality assurance (QA), development engineers who write specifications or implement resource strings (text seen on the UI), translators who translate (localize) from English as the source language, as well as anyone working on documentation that refers to the OpenOffice.org UI.

---

## 2. Consistent Text and Terminology

It is vital that users do not get lost when traversing the user interface. Text is just as important as colors and shapes in guiding users to the right elements to perform desired tasks, even more so for blind users. “Text” refers to any strings, words, phrases

or sentences, whereas “terms” or “terminology” refers to individual units of meaning, which may or may not be limited to one word. Some example terms are “data source”, “plug-in” and “paste special”, but also key verbs like “select”, “double-click” and “mark”. Whereas an example of text containing terms could be, “Connect to the data source”. For information on specific terms, go to the Tricky Terminology chapter.

## Use Existing Text and Terminology

Most importantly, all text and terminology already used in OpenOffice.org must be used as the basis for creating additional text or introducing new terminology. That means, for example, that if a word is already used in a particular meaning in OpenOffice.org, you cannot introduce a synonym based on a new feature specification, despite what the competition might write. That would cause confusion not only for users, but also for translators and technical writers. Thus, the feature specification has to be corrected to use the existing terminology.

## No Multiple Meanings for Terms

One term must have one meaning (concept) and, likewise, one meaning must always be identified by the same term. If a usability test shows that a particular term is not understood by the users, you cannot just change that one instance of the term, but must change every instance in the whole office suite. Only then will the terminology be consistent and usable. That is why changing already existing UI text is never done without thorough research of the text in OpenOffice.org. In the same way, text used in new features must always correspond to existing text and terminology or the designer must consciously initiate a change in all old text to correspond with the new text.

## Write for Users With Basic Knowledge

If you are reading this guide, then your task at hand is likely to ensure that the English and German text used in OpenOffice.org is grammatically correct and is molded according to the terminology and style appropriate for an office suite. Your target audience is end users of office suite software. Generally, assume those end users are familiar with basic functionality of computers and office suites and that they have at least a high school education. In particular parts of OpenOffice.org that are more technical or advanced, such as macro programming or the database application, it is safe to assume that the users are more advanced and can understand a more technical vocabulary.

## How to Make Decisions on UI Text

The text on the user interface must be consistent, coherent and clear, especially when bridging from one element to another, such as from menu to dialog, or from button to dialog. How can you decide what text or term to use? Look at the existing OpenOffice.org UI, research OpenOffice.org terminology already used, compare with other products and sources, etc. It is important that the new features coming in fit with the UI around them.

### Have Dialog Titles Match Menu Items

When choosing the text for the title bars of dialogs, the decision of how to name them is easy: Keep the dialog name the same as the menu item that is used to call the dialog.

*Correct:* menu item “Bullets and Numbering”, dialog title “Bullets and Numbering”

*Incorrect:* menu item “Printer Settings”, dialog title “Printer Setup”

If a dialog can be called by several different menu items or buttons, the menu or button labels and the title of the dialog should at least contain overlapping terminology and thus be obviously related.

*Correct:* main menu “Format - Cells” or context menu “Format Cells”, dialog title “Format Cells”

### Think About Priority

In terms of terminological priority (on a scale from high to low), the main menu items are “high”, followed by submenu items and text in main dialogs that are directly opened by choosing a menu as “medium-high”. The approx. 800 error messages in OpenOffice.org range from “medium-high” to “low” depending on how often they appear and how important the situation is in which they appear. Keep the priority level in mind when revising existing text or planning time to create or review new text. A basic rule of thumb is: If you come into contact with a certain menu, dialog or error message every day when doing simple tasks with OpenOffice.org, then it is high priority. If it is a menu, dialog or error message might only appear to a handful of users, for example because it deals with a seldom-used feature, then it is low priority.

---

## 3. Tools for Text and Terminology

### Terminologies and Glossaries

#### OpenOffice.org Developer's Guide

You can find a glossary (=term + definition) for the OpenOffice.org Developer's Guide on the OpenOffice.org website:

- [api.openoffice.org/docs/DevelopersGuide/Glossary/Glossary.htm](http://api.openoffice.org/docs/DevelopersGuide/Glossary/Glossary.htm)

## OpenOffice.org English-German Terminology

There is currently no English-German terminology list (=English term matched with German term) for the OpenOffice.org UI.

In the meantime, Elizabeth Matthis worked with the German OpenOffice.org project volunteer Wolfgang Henderkes to provide the German documentation volunteers with a terminology to facilitate their writing and translation. Wolfgang's *HuT-eSaurus*, can be accessed via his website:

- [www.it-know-how.net/esaurus.htm](http://www.it-know-how.net/esaurus.htm)

You can download the German-English terminology in the *HuT-eSaurus*, which is not exclusive to OpenOffice.org and constantly being updated, directly at the following address:

- [www.hut-online.de/transfer/sw0143dl.zip](http://www.hut-online.de/transfer/sw0143dl.zip)

## SAP Terminology

The German company SAP also produces German and English UIs and documentation in parallel. They have made some short terminology lists publicly available:

- [www.sapdesignguild.org/resources/References/index.htm](http://www.sapdesignguild.org/resources/References/index.htm)

## Gnome Glossaries and Other Resources

GNOME is also targeted toward end-users, so the terminology is akin to that which is used in OpenOffice.org or could be adopted for use in OpenOffice.org when new terms are being researched. Be careful, however: GNOME terminology is not the same as OpenOffice.org, for historical as well as market reasons.

- [developer.gnome.org/documents/style-guide/index.html](http://developer.gnome.org/documents/style-guide/index.html)
- [usability.gnome.org/hig/2.0/](http://usability.gnome.org/hig/2.0/)
- [usability.gnome.org/hig/2.0/hig-2.0.pdf](http://usability.gnome.org/hig/2.0/hig-2.0.pdf)
- [gnome.org](http://gnome.org)

## Internet Dictionaries

- [www.computer-dictionary-online.org](http://www.computer-dictionary-online.org)
- [www.techdictionary.com/](http://www.techdictionary.com/)

- [www.webopedia.com/](http://www.webopedia.com/)
  - [www.acronymfinder.com/](http://www.acronymfinder.com/)
  - [whatis.techtarget.com/](http://whatis.techtarget.com/)
  - [foldoc.doc.ic.ac.uk/](http://foldoc.doc.ic.ac.uk/)
  - [www.computeruser.com/resources/dictionary/](http://www.computeruser.com/resources/dictionary/)
  - [www.m-w.com](http://www.m-w.com)
- 

## 4. Tricky Terminology

---

Consistent use of terminology is important for several reasons. Not only does it make the UI easier to understand and support a particular product image, but it is also extremely important when the text is to be the source language for translation into other languages. The spelling conventions throughout the UI must be the same and the same meaning should be attributed to the same word or phrase wherever it is used.

### Specific Terms

Only important or problematic words are presented here. As these words appear very frequently, it is especially important that they are used consistently and that no synonyms or variations are used instead.

#### “Mark” Versus “Select”

“Select” is used to refer to the selection of objects/items, for example, paragraphs or characters which are then deleted, moved, copied, given a new font, etc. It is also used when selecting options from list/combo box drop-down lists and for radio buttons.

To discuss the use of check boxes, use the verbs “mark” and “unmark” or the adjectives “marked” and “unmarked”. Writing “mark” is also in the interest of using internationally accepted English, because it side-steps the usage of “tick” versus “check”. “Tick” is unacceptable in US English, whereas “check” is sometimes considered incorrect in UK English, where “tick” is more common. In the past, other words like “select”, “check”, and “activate” were used inconsistently alongside “mark”. Our goal is to avoid using different words to refer to the same function.

In German, always use “auswählen” whenever “select” is used in English. “Markieren” is not correct either.



*Correct:* Wählen Sie den Text aus.

*Incorrect:* Selektieren Sie den Text.

## When to “Choose”

Use “choose” to:

1. Open a menu.
2. Choose a menu item.

The German translation of “choose” is “auswählen”, the same as for “select”.



## How to “Click”

Use this verb to describe the action used to activate icons, various buttons and other functions on the user interface. Avoid using “click on”. (However, in German always use “auf”: &ldqu;Klicken Sie auf OK”) Variations of click are as follows:



<b>Term</b>	<b>Definition</b>
Click	To quickly press and release the left mouse button one time, without moving the pointer.
Double-click	To quickly press and release the left mouse button twice in rapid succession.
Left-click	The same action as <i>click</i> ; to quickly press and release the left mouse button one time, without moving the pointer. Use the term <i>left-click</i> to distinguish the action from <i>right-click</i> when they appear in the same paragraph.
Long-click	To press and hold down the left mouse button for about three seconds, then release it. For example, you use a <i>long-click</i> to open a floating toolbar from the main toolbar.
Right-click	To quickly press and release the right mouse button one time. For example, you use a <i>right-click</i> to open context menus.
Single-click	The same action as <i>click</i> ; to quickly press and release the left mouse button one time, without moving the pointer. Use the term <i>single-click</i> to distinguish the action from <i>double-click</i> when they appear in the same paragraph.



Left-handers please note that right mouse button and left mouse button are switched for your mouse if you have a left-mouse, or if you have modified your mouse settings.

### “Want” Versus “Wish”

Use the verb “want” instead of “wish”. For example, “You can save the document in a different format if you want.” (even better would be “...if required” or “...if necessary”)

### “Dialog” Versus “Dialog Box”

Instead of using “dialog box” as is done in many other programs, use only the term “dialog” in OpenOffice.org. There are three reasons why this was decided:

1. “Box” is an unnecessary extra since there are no conflicting cases of dialogs without boxes.
2. This style has been used consistently throughout the user interface and end-user documentation up till now so changing it would incur huge rework costs with little benefit to the users.
3. GNOME, also only uses “dialog”.

### “Drag and Drop”

When used as a modifier, write “drag-and-drop”. For example, “the drag-and-drop operation”.

You can use “drag” as a stand-alone verb. The action of dragging includes dropping the object so you do not need to add “drop”.

*Correct:* Drag the frame to another place on the page.

*Incorrect:* You can drag and drop the frame to another location on the page.

---

## 5. Consistent Writing

---

A positive user experience depends greatly on the consistent use of language, a consistent writing style and consistent capitalization of the user interface elements. The resulting product style conveys the assurance of serious, reliable, professionally-designed software.

## American Versus British English

Use US English (the standard in the United States of America) rules of writing and spelling in OpenOffice.org. Refer to the *Merriam Webster Dictionary* for meanings and spellings in general. The dictionary is available online for free at: [www.m-w.com](http://www.m-w.com).

## Writing Style

The OpenOffice.org software is for end-users, that is, anyone who would like to type a letter or any other everyday task in countless offices and homes. The audience may include, but certainly does not target technical people like system administrators or engineers. That means that the writing style should be in a non-technical, every-day style. The goal is to have as many people as possible comfortably using OpenOffice.org. Therefore, the UI has to be easy to understand and user-friendly. Two other important aspects that writers need to be conscious of are that the English text is the source text used for translations into all other languages, and that a large number of people using the English version of OpenOffice.org are probably not native speakers of English. Therefore, be sure to read the additional special writing style guidelines in the Keep Localization in Mind section of this guide.

## General Style

- Do not use marketing language.
- When writing sentences, keep them short and simple and use correct punctuation.
- Do not be witty or try to be humorous.
- Avoid idiomatic phrases or slang.

## Write Complete Company or Product Names

For legal reasons, always write out full company names, especially when you mention another company. This is also good writing style just to make sure that all readers understand exactly what you mean, as not all abbreviations are universal.

*Correct:* Microsoft Word, Sun Microsystems

*Incorrect:* MS Word, SMI (or only Sun)

A similar rule applies to products. Unless there is a commonly used short form that is usually even more well-known than the full name, for example “SQL” or “JDBC”, write the complete name. This is also a marketing issue.

If the context is a wizard or other series of dialogs that the user will see in a row, at

least write out the whole name the first time and put the short form in parentheses behind it. Explaining the short form in parentheses should be used only when necessary to not annoy users by constantly repeating a long name.

*Correct:* Java runtime environments (JRE) already installed: xxxxxx

(that was the first mention; then the error messages triggered by this dialog only say “Different JRE Required” in the title)

*Incorrect:* JRE already installed: xxxxxx

Otherwise, just always write out the correct, complete name. You may know what the short form means, but a user or translator may not. Sometimes, however, there are official short forms.

## No Abbreviations

Most of the time, abbreviations should not be used on the UI. Even common abbreviations such as “esp.” (for “especially”) and “no.” (for “number”) are not universally recognized by all speakers of English and make the UI design look hacked. In addition, some abbreviations can be interpreted in multiple ways, for example, “min.” can be “minimum” or “minutes”. However, certain abbreviations are allowed if they are commonly used in office suites, for example KB, MB, etc.

## Avoid “Please”

Use “please” only rarely, if ever. Sometimes lengthy instructions or error messages might need a touch of politeness added to one of several sentences, but generally, do not use “please” in English. If you still feel the need, at least avoid using “please” more than once in one text or group of sentences. For example, you could use “please” in the initial instructions on the first page of a wizard, but then not on any of the other pages. Simple imperative sentences are always preferred.

*Correct:* Click 'Shut Down'.

*Incorrect:* Please click 'Shut Down'.

(Note: Contrary to the English rule, “bitte” is used in German most of the time. Add it to the German text even when it is not used in English. (different culture = different conventions)



## No Contractions

Never use contractions such as “can't” or “it's”. The only exception to this rule is the

menu item “What's This?” (and perhaps that should be revised). The OpenOffice.org style is clear and professional, not chummy or informal, which is the feeling many speakers of English get when reading contractions. In addition, they can be misunderstood by both translators and users so it is best to avoid any ambiguity by writing out the full text.

*Correct:* do not

*Incorrect:* don't

## No Spaces Before Punctuation

Although it may be common in other languages like French, there are no spaces before punctuation in English. Nowadays spaces are sometimes used in email communication to allow particular punctuation to stand out, but that is an informal style that should not be found in a professional software product. It is **not** more helpful to the reader when you leave spaces. It is simply an annoying error. Therefore, do not insert spaces before punctuation marks.

*Correct:* An error has occurred. Select one of the following options:

*Incorrect:* An error has occurred . Select one of the following options :

## Do Not Use the Em dash

Unlike its cousin, the hyphen (for example, in the modifier “look-a-like”), dashes are not to be used when writing UI text. Often found in informal writing, the em dash functions as a “super-comma” or set of super-commas to set apart elements of the text, especially when those elements have additional forms of punctuation or the writer does not know the correct punctuation to use. However, UI text should never contain thoughts or be so complicated that an em dash would be necessary to separate otherwise hard-to-understand parts of a sentence. If the text you are contemplating for your design uses an em dash, rewrite it using other punctuation or separate it into two or more sentences.

*Correct:* Click the button. The dialog appears.

*Incorrect:* Click the button—the dialog appears.

## No Space Before Ellipsis (Three Dots)

menu items that cause a dialog to open are followed by three dots, also called “ellipsis”. Never put spaces between text and the dots. The extra space is not helpful and is incorrect in English as well as German.

*Correct:* Browse...

*Incorrect:* Browse ...

## Use Spaces Between Text and Arrows on Buttons

In a dialog or wizard, some buttons have additional characters aside from the text. Leave one space between the character and the text. For example, there must be a space between "<" and "Back" and between "Next" and ">".

*Correct:* < Back

*Incorrect:* <Back

## Avoid Ampersands (&)

Do not use the ampersand symbol "&". It makes those texts that employ it stand out more than others, for example when used on a menu, by disrupting the overall look. Due to its exclusive nature, the ampersand also causes uncertainty in localization concerning whether or not it signifies that the text is a proper name and therefore not to be translated.

*Correct:* Boys and Girls

*Incorrect:* Boys & Girls

Note: As of OpenOffice.org 8, the ampersand is still used in a few cases. These should be cleaned up as time goes by.

## Do Not Use Exclamation Points (!)

Do not use exclamation marks, even if you are writing a warning or error message. In English, exclamation points show strong emotion or emphasis and are used much less frequently than in German. Users are meant to feel comfortable using OpenOffice.org, not frightened. There is no error or problem that warrants text punctuated by an exclamation point and therefore showing or eliciting strong emotions.

In German, exclamation points can be used, but please do so with an attitude of "British understatement" to curb the German habit of using exclamation points much too often.



## Have Real Messages in Error Messages

End users should not have to be confronted with messages like “Error found: Abort now”. If we display a message for our users, there has to be information in it that says what has occurred and give the user instructions on what steps to take next.

Error messages, warnings, confirmation dialogs and other message boxes (sometimes also called “alerts”) must be written in complete sentences. They should not have any technical information in them, unless the message pertains to a part of OpenOffice.org tailored to a more advanced technical audience.

*Correct:* The file already exists. Do you want to overwrite it?

*Incorrect:* The file already exists. Overwrite?

For more information on writing error messages and other alerts, refer to the guidelines:

- [java.sun.com/products/jlf/ed2/book/HIG.Dialogs5.html](http://java.sun.com/products/jlf/ed2/book/HIG.Dialogs5.html)
- [Alerts & Warnings](#)

## Use Present Tense and Imperative

Use the present tense wherever possible. Avoid all other tenses, except where you need to specify events in terms of the past or the future. Use imperative verbs (without “you”, for example “To assign ..., click 'OK' .” and not “To assign ..., you click 'OK'.”) for instructions and most check box labels.

*Correct:* Wrap text automatically

*Incorrect:* Wrapping text automatically

In German, do **not** use the imperative because it requires either the informal “du” form or the formal “Sie” form. In the German version of OpenOffice.org, we always use the formal “Sie” form of address. In the imperative, it is too cumbersome to write the formal imperative. Instead all German text corresponding to English imperatives is written in the infinitive.



*Correct:* Einstellungen des übergeordneten Objektes verwenden

*Incorrect:* Verwenden Sie die Einstellungen des übergeordneten Objektes

*Incorrect:* Verwende die Einstellungen des übergeordneten Objektes

## Avoid “Thing's” Possessive Forms

Avoid using the Saxon genitive ’s for possessive cases, especially for inanimate objects. Use a possessive sentence structure with “of” instead.

*Correct:* On the left side of the screen.

*Incorrect:* On the screen's left side.

## Do Not Use “May”

As the modal verb “may” is ambiguous, even to native-speakers of English, in that it can express permission to do something, a possibility (“might”) or a real ability (“can”), use the words “might” or “can” instead. This will avoid ambiguity in the English text which is not only better for the readers, but also for the translators.

## Prepositions

There were several different people working on the OpenOffice.org English version in the past and not all of them were native-speakers of English. Due to this and differences in preposition usage throughout the English-speaking world, consistent use of prepositions has always been a problem. We need to present the readers with a consistent style, so certain prepositions have been defined for certain contexts. These are the most common cases:

- in a window
- in a document
- on the task bar
- on a floating toolbar
- on a tab page
- from the menu (or “on” when the context does not fit “from”)
- on the user interface

## Use Single Quotation Marks, Never Double

When quoting other UI text or file or document names (or anything similar that would be put in quotation marks in normal prose), always use single quotation marks, not double ones. This is due to technical issues in resource files: For some code the double quotation marks have to be “escaped” by preceding them with a slash, but the slash causes confusion in localization so it is not worth the hassle. Even if more modern development environments make the escaping unnecessary, the writing style using single quotation marks is already firmly entrenched in OpenOffice.org, so the style must continue to remain consistent.

*Correct:* Click 'Install' to continue the installation.

*Incorrect:* Click "Install" to continue the installation.

## Use Quotation Marks When Quoting UI Text

It has already been mentioned that use of single quotation marks is the correct way to highlight UI strings being quoted on the UI. Examples of UI text that often gets quoted are button labels or check box names that are referred to within directions in a dialog or message box.

Highlighting with single quotation marks is employed rather than only noting the label or name as it is seen and capitalized onscreen. This makes the text more legible, especially for quick readers scanning to see the words that directly indicate UI elements they have to click or take action upon.

### Quoting UI Text

*Correct:* Click 'Install' to continue the installation.

*Incorrect:* Click Install to continue the installation.

### Quoting Document Names

*Correct:* The document will be saved under the name '%DOCNAME'.

*Incorrect:* The document will be saved under the name %DOCNAME.

### Quoting UI Text in German and Other Languages

The single quotation marks in OpenOffice.org signal to the translator that the text within the quotation marks is a UI text being quoted and that they should deal with it accordingly. The style guide for the respective languages should dictate whether or not the single quotation marks are kept, removed or altered for that language. (Just because the quotation marks, or any other form of highlighting or punctuation for that matter, are used in the English source language, does not mean they have to be used in the target languages.)

Many languages do not differentiate between title and sentence capitalization. The text can even be interpreted by the reader as a spelling or grammar error when a quoted UI text appears without quotation marks and is therefore not immediately identifiable as a quoted UI element, for example, a button label. When only capitalization is used to "highlight" a quote from the UI, it can force the translation to be grammatically incorrect. In Spanish, a whole title quoted within a sentence cannot be capitalized, so the compromise is to capitalize the first word of the title. The rest of the title blends with the rest of the sentence and makes reading the text awkward if not incomprehensible altogether. (A title would be emphasized in bold or cursive in a text written directly in Spanish.) In German, verbs are not capitalized within a sentence, for

example, but verbs used on UI buttons are capitalized. Therefore, the single quotation marks are used in German UI text, just like in English:

*Correct:* Klicken Sie auf 'Abbrechen'.

*Incorrect:* Klicken Sie auf Abbrechen.



## Intuitive Shorthand

OpenOffice.org uses its own style of intuitive shorthand to quickly and effectively indicate menu selection and other action paths that the user needs to follow. The OpenOffice.org shorthand to show the user which steps to take looks like this:

Choose File - Print - Properties - Paper

Each UI element is separated from the next one by a space, a simple dash and another space. Although descriptions such as this are usually only found in the documentation, it can sometimes be necessary to explain steps in UI messages as well. Furthermore, descriptions of steps to locate elements on the UI are required in issues, specifications or localization guides, so using the same shorthand as the documentation can be helpful. If the intuitive shorthand in the documentation were to change, the UI would have to follow suit. Contact the OpenOffice.org technical writers if you are not sure.

---

## 6. Keep Localization in Mind

As the English text serves as the source language for localization into up to 50 other languages, the text must be chosen very carefully. This section points out specific important issues to take into account when creating UI text for OpenOffice.org.

### Include Variables (Placeholders) in Text

Include variables in the text you create. Variables are sometimes also called “placeholders” by developers because the variable holds the place for the text that will replace it on the UI at runtime. The variable is only visible in the resource string (in the resource file in the code) and not on the UI. (The preferred term, “variables”, will be used in this guide because “placeholders” is usually used to refer to real text seen on the UI which the user then clicks to overwrite, such as “<fill in your name here>”.)

Do not leave text phrases open-ended or separate two parts of a text into separate resource strings. For developers it is easy to hard-code the additional text, such as a URL, a number or a file name, that will be inserted at runtime. They do not have to include it in the resource string and may think they are saving time or money in doing so. However, if something is going to be inserted at runtime, the translator must know it, and should be told what the text will be. Doing so allows the translation to have

correct grammar and punctuation, which would otherwise not be possible. Including variables also allows for easier reviewing of the original text in English, too.

*Correct:* The document will be saved under the name '%DOCNAME'.

*Incorrect:* The document will be saved under the name

## Keep Variables Consistent

The standard OpenOffice.org form for variables is: "%CAPITALLETTERS".

Due to the long history of the development of OpenOffice.org and the fact that only within the last 2-3 years was there an effort to limit the variable style to one standard, there are currently many forms used in the OpenOffice.org code, for example:

### Prefix, Suffix

"&", ";"

"%", "%"

"%", None

"\$(", ")"

"\$", "\$"

"\${", "}"

"#", "#"

"(\$", ")"

"\$[" "]"

Unfortunately, due to different programming languages and development environments being used, some symbols are prescribed or restricted. However, it should be possible in most cases to use the standard style as long as the developer knows the style before starting to code. A concerted effort to convert ("fix") the other variables to the standard form will greatly reduce the amount of time spent explaining the variables to the localization providers at every turn.

## Reuse Existing Variables

Never create new variables when there are already variables defined to achieve the same result. When writing specifications, be sure to consult LingTool or this list first. The most commonly occurring variables are:

%PRODUCTNAME

Will be replaced by OpenOffice.org, StarOffice, etc.

%PRODUCTVERSION

Will be replaced by the latest version number 8, 2.0, etc.

%MODULENAME

Will be replaced by “Writer”, “Calc”, “Math”, “Basic”, “Impress”, “Draw” depending on the context.

%ORGANIZATIONNAME

In the error report tool, replaced by “Sun Microsystems” or “OpenOffice.org”.

%PRODUCTPATCHNAME

Stands for “Product Update 2” or “Product Update 3” and later numbers depending on version.

%PRODUCTEXTENSION

Will be replaced during installation with EA1, EA2, or BETA (not used in final version=empty)

%PRODUCTXMLFILEFORMATNAME

In list box of filters or type of document, this will be replaced by “StarOffice”, “OpenOffice”, etc.

%PRODUCTXMLFILEFORMATVERSION

In list box of filters or type of document, this will be replaced by “6.0”, “1.0”, etc

### Create Variables According to Guidelines

Good programming style says to use self-explanatory names for variables and this is in the interest of localization, too. Additionally, if technically possible, good variables:

- never use punctuation marks,
- never use lowercase words,
- never use real words (instead: push them together or shorten them),
- never use #.

The reasoning behind these rules is that translators who localize OpenOffice.org come and go and are not always professionals (for example, OpenOffice.org volunteers). The best solution is therefore to maintain clarity and consistency at the engineering-end of the process. To understand the problems translators can have identifying variables as something not to translate, remember the following:

1) Translators are not developers and do not necessarily know what a variable is. They could think it is a typo. “I’ll just leave out the funny character and my translation is at least correct, even if the original is buggy.”

2) A variable very often looks just like a real word with a typing or character encoding

error in it. “Is it a bug to report or perhaps some developer thingy? Hmmmmm...”

3) Some UI strings contain placeholders or field labels such as “<number>” that have to be translated, for example in German “<Nummer>”. So a variable written “<path>” would look a heck of a lot like something to translate, for example in German “<Pfad>”, but, oops!, the code is broken.

4) If machine translation is being used, the variables have to be fed into the machine in advance so it knows not to translate or change them. Keeping track of all different forms of variables for the machine is difficult, especially when new variables get invented all the time.

5) “#” is the symbol for “number” in the USA. This is sometimes used as a button label on a UI. In that case, it has to be localized, for example in German “Nummer”. So do not think that a symbol you choose for a variable is going to be obviously non-text, just because you wouldn't use it in your language. A variable such as “#1” could mistakenly be translated into German as “Nummer 1”.

6) Sometimes it is not clear where the variable starts and ends, which symbol is part of the variable and which one is regular punctuation in the text, for example, brackets <>[], parentheses () or apostrophes ' can all be confusing for translators.

7) A new translator will generally assume that OpenOffice.org/OOo is one product, and that therefore there is one style for text and variables. [one product = one programming language = one style]

## No Phrases Ending in Prepositions

Do not end phrases with prepositions, for example, fixed text introducing a group of check boxes. Prepositions have to be declined in many languages, including German, and they must therefore be placed with the noun that follows them.



English	German
<i>Correct:</i> Send outline <ul style="list-style-type: none"><li>• To presentation</li><li>• To clipboard</li><li>• As attachment</li></ul>	<i>Correct:</i> Gliederung versenden <ul style="list-style-type: none"><li>• Zur Präsentation</li><li>• Zur Zwischenablage</li><li>• Als Anhang</li></ul>
<i>Incorrect:</i> Send outline to	<i>Incorrect:</i> Gliederung versenden zur

- Presentation
- Clipboard
- Attachment

- Präsentation
  - Zwischenablage
  - Anhang
- 

## No Non-international Symbols Like "#"

Although localization specialists should be able to identify symbols commonly used in English, they may not be able to understand the context and find the applicable localization. Often symbols are used when there is a lack of space, which is just asking for trouble because the localization is guaranteed to be longer, generally having to use words to localize the symbol used in English.

Not all symbols, such as "#", are recognized as having the same meaning all over the world---even among native-speakers of English. For example, "#" to mean "number" is very commonly used in the USA, but has been reported as hardly known in the UK. As it is hard to know which symbols are universally recognized, it is best to use words instead of symbols.

## Be Aware of Text Length

English text is much shorter than most of the localizations which correspond to it. In fact, the localized text can be three times as long. As a rule of thumb, check the Brazilian Portuguese translation (by searching in LingTool) to see how many characters and spaces will have to be visible on the button or in the space of the dialog. Sometimes, however, Italian or French can present the longest translation. In OpenOffice.org languages, Finnish seems to do a good job of competing with Brazilian Portuguese for longest text. In the end, quality testing of localized builds is the only way to find any imperfect UI text that would call for a spatial adjustment in the design, so just do your best.

Until OpenOffice.org gets a layout manager which would automatically adjust the space needed to accommodate the text in different languages, the UI designer must leave enough room on the screen to allow space for the longest translation. That is also why most of the text is positioned above a control, such as a text box, or after a control, such as a check box.

## No Spaces Before or After Text

Do not start a text with a space or add an additional space after the text. Unfortunately, this was done in the past by developers when implementing the text. They added

spaces to the text as a way of providing formatting or spacing of text units on the UI. Not only is that not good coding, but it will cause errors in the translation database as well as begs for errors in localizations, where the translator will usually not include the extra spaces and therefore lose the “formatting” on the UI. If the UI design prescribes indentation or other spacing of text, then it must be accounted for in the code, not as part of the text string.

---

## 7. Capitalization

The consistency of capitalization gives the software user interface a professional, mature look and also provides consistent and sometimes valuable input for translators working on the localization.

### Basic Rules

The basic rules regarding title and sentence capitalization summarized below only apply to English, not to German.

In German, use the regular rules of capitalization as you would in any normal text. Start with a capital letter as if it were a sentence, even if it is only a phrase, such as a check box label. Following this same concept, all button labels also start with a capital letter, even if they are a verb.



### Title Capitalization

In the title capitalization style, capitalize every word except articles ("a," "an," and "the"), coordinating conjunctions (for example, "and," "or," "but," "so," "yet," and "nor"), and prepositions with fewer than four letters ("in").

*Correct:* Coding Without Your Eyes

*Incorrect:* Coding without your Eyes

In title capitalization, always capitalize the first and last words, even if they are prepositions or any other word that would normally be lowercase within a title.

*Correct:* Save As

*Incorrect:* Save as

### Sentence Capitalization

In sentence capitalization, only capitalize the first word. Exceptions are proper nouns, abbreviations, or acronyms that are always capitalized, as well as quotations of UI text.

*Correct:* Click 'Decline' if you do not accept the terms of the License Agreement.

*Incorrect:* Click 'decline' if you do not Accept the Terms of the license Agreement.

In the example above, “License Agreement” is considered a proper noun because it is defined in the agreement and always referred to in capital letters to make it clear that *that* specific legal definition is legally binding for *that* agreement.

## User Interface Elements

Most elements on the user interface, sometimes also called “controls”, have text labels, tooltips or accessibility descriptions (read to visually impaired users via assistive technology, screen readers, for example) associated with them.

The purpose of the following table detailing how various UI elements are capitalized is to promote consistency throughout the product in dialogs, menus, etc. and consequently in any related documentation that quotes the UI text.

The names in the second column below correspond to the names developers see in the source code (resource files) as well as to the “Types” noted in the translation database “LingTool”.

### *Table 1 Capitalization of UI Strings According to String Type*

There may be exceptions to these capitalization rules in older parts of the software that were designed before the style guidelines were written. In such cases, keep that already existing style consistent until an opportunity arises (for example, a new feature requires the dialog to be redesigned) to revise the strings to adhere to the style in these guidelines. The goal is to have all inconsistent strings replaced or deleted as time goes by.

A similar table is publicly available for OOo:

- [ui.openoffice.org/files/documents/59/227/EnglishGUIstyle.sxw](http://ui.openoffice.org/files/documents/59/227/EnglishGUIstyle.sxw)

### One Text Only for Menu, Tooltip and Customize Dialog

As of OpenOffice.org 8, a central file was created so that only one resource text (string) would exist for all main menu items, the tooltips of the corresponding icons, and the corresponding names in the list in the Customize Toolbar dialog (1=3). Therefore, changing the text of a menu item in the central file also changes the corresponding tooltip text and entry in the Customize dialog and vice versa. This is in the interest of maintaining terminological consistency, in English and all other languages. There are very few exceptions (maybe 2) where the tooltip has to have a slightly different text

than the menu item. These exceptions have to be supported by extra mechanisms in the code. Exceptions to this 1=3 handling are only allowed when absolutely necessary, due to the extra maintenance effort.

---

## 8. Text for Wizards

Quoted from sections 5.3 and 5.5 in the specification:  
[specs.openoffice.org/installation/wizards/Wizards\\_NewConcept.sxw](https://specs.openoffice.org/installation/wizards/Wizards_NewConcept.sxw)

### Title Bar

The title of the title bar represents the name of the wizard. Use title capitalization style. It remains the same over all pages of the wizard.

### Left Pane (Roadmap)

The roadmap steps may be expanded over two lines, but they should be kept as short and precise as possible. Verb phrases are preferred, but noun phrases can be used instead. You should use the same grammatical type consistently for the whole roadmap.

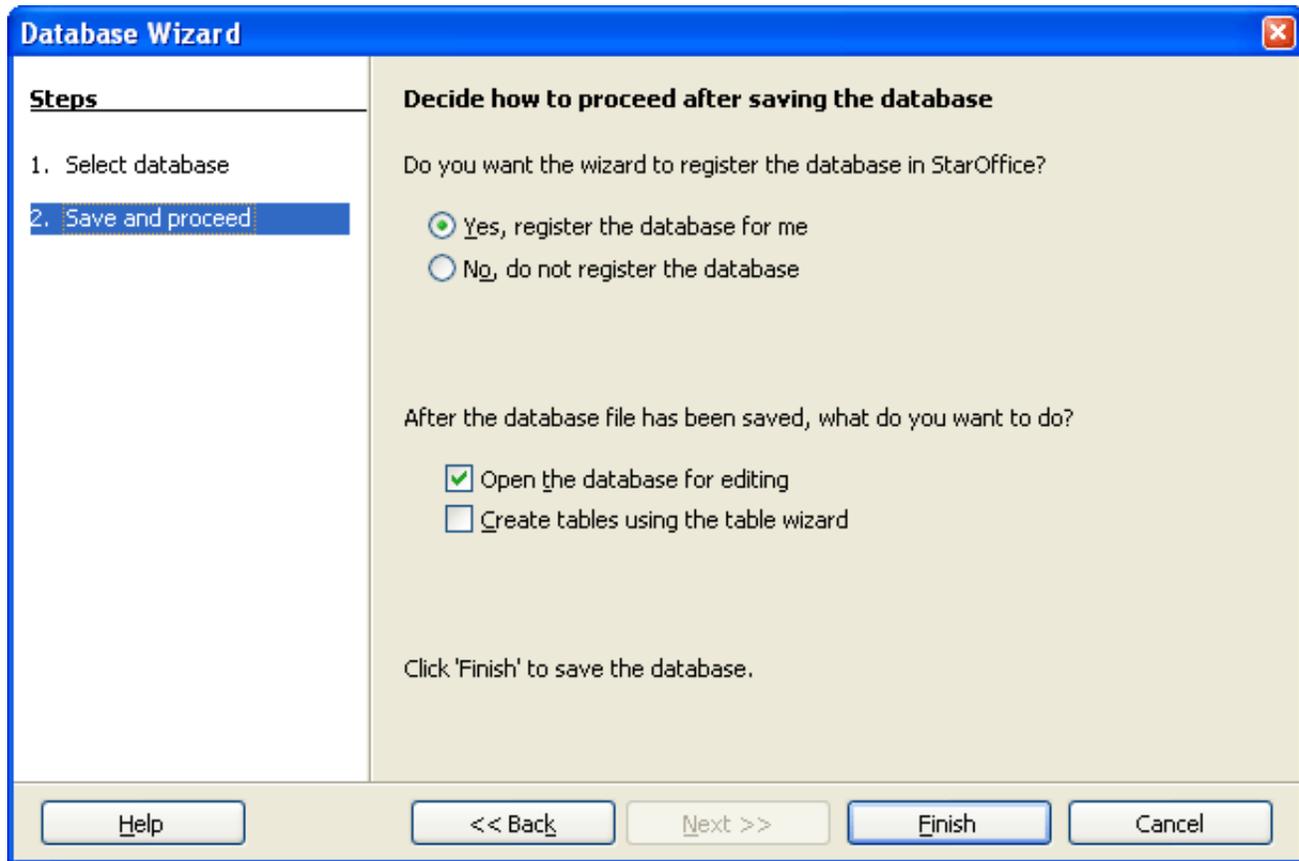
Each roadmap step must correspond to the heading of the page it calls in the right pane. That means that the words of the roadmap step are used in the heading, if at all possible. Use sentence capitalization but do not put a period or other punctuation at the end even if it is sentence. The sentence capitalization makes it possible to have shorter as well as longer text for the roadmap, depending on the purpose of the wizard. If title capitalization were prescribed, then longer text, for example a sentence, would look ridiculous and not be as legible. For the same reason, we leave the period out: Only full sentences should end with sentence punctuation, but most of the time the roadmap will not contain sentences. Therefore by never using the period, we avoid inconsistency in the overall look for all wizards.

### Right Pane

The heading of the page in the right pane must correspond to the roadmap entry that calls that page. The heading should be in the imperative (not a question). The imperative form introduces the information dealt with on the active dialog page by directing the user to do what is to be done on that page. For example “Enter name, address and telephone number” or “Select asset types”.

Use sentence capitalization but do not put a period or other punctuation at the end of the sentence, since this is a heading. The sentence capitalization makes it possible to have shorter as well as longer text at the top of a dialog or wizard page. If title capitalization were prescribed, then longer text, for example a sentence, would look ridiculous and not be as legible. For the same reason, we leave the period out: Only full sentences should end with sentence punctuation, but sometime headings are not full

sentences. Therefore by never using the period, we avoid inconsistency in the overall look.



## Mnemonics

Make sure the mnemonics (hotkeys) are defined for every control label (not for text without a control, such as group headings in dialogs). Mnemonics are necessary to make the controls accessible via keyboard instead of only via mouse. This is very important because it enables the software to comply with accessibility requirements in “Section 508”.

Mnemonics only have to be set manually in English strings, not German or other foreign languages, as there is a macro that generates the mnemonics at runtime for other languages. In Asian languages, the English mnemonic will automatically be set in parentheses following the Asian string. The macro could also be used to create mnemonics for English, as is indeed the case if they are accidentally not



set by hand, but because the English mnemonics directly affect other languages, they have to be carefully chosen and not generated randomly.

Use the mockup of the dialog or menu in the design specification to assist you in finding which letters to use. The mnemonic in the string resource file is denoted by a tilde in front of the letter that will be underscored onscreen.

---

## 9. Appendix A

### Checklist

This checklist is meant to provide quick input for people creating UI text. The points below are arranged starting with the most important ones and ending with the lesser important ones. To learn more about why UI text must conform to the points on the checklist, read the corresponding sections in this guide.

The checklist does not include every rule you should follow when creating text, only the most critical and obvious ones. Additional information can be found in the various sections of this guide, including examples and specific details.

- Write short and simple sentences without special typographic elements
- *no m-dashes, avoid semi-colons, no exclamation points (!)*
- Use consistent text and terminology
- *select=auswählen, Click...=Klicken Sie auf...*
- Make dialog titles match menu items
- Use title and sentence capitalization consistently for the UI element labels
- Use American English
- No abbreviations
- *write "number" instead of "no.", "for example" instead of "e.g."*
- No contractions
- *use "cannot" instead of "can't" or "do not" instead of "don't"*
- No phrases ending in prepositions
- *write "Send outline" instead of "Send outline to"*

- Use 'single' quotation marks, never &ldqdouble"
- Write "dialog", not &ldqdialog box"
- Use single quotation marks when quoting UI text, document names
- No space before ellipsis (...)
- Keep variables consistent and according to the OpenOffice.org style
- Have real information and complete sentences in error messages
- No spaces before or after text
- Do not use "may"
- No spaces before punctuation
- Include variables (placeholders) in text
- Avoid "please"
- Do not write "wish"
- Avoid "thing's" possessive forms
- *write "The title of the dialog" instead of "The dialog's title"*
- Use consistent prepositions, for example:
  - *on a floating toolbar*
  - *on a tab page*
  - *from the menu (or "on" when the context does not fit "from")*
  - *on the user interface*
- No non-international symbols like "#"
- Always write full competitor's company or product name
- *write "Microsoft Word" instead of "MS Word"*

## 10. Version History

<b>Version</b>	<b>Date</b>	<b>Who</b>	<b>What</b>
2.19	16.12.05	Christian Jansen	Created OpenOffice.org Version