Documentation Style Guide

SUSE and openSUSE

Version 2

$Revision: 39 $

Build Date: October 01, 2010

Legal Notice

Disclaimer All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither Novell, Inc., SUSE LINUX Products GmbH, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

Contents

This document contains SUSE-specific style and layout issues and writing guidelines. It identifies and describes several issues to standardize the style of SUSE and openSUSE documentation. This document also contains a reference part for the XML markup used by the SUSE editors and intended for openSUSE projects, such as *Lessons for Lizards*.

This document refers to the following documents:

- *The Chicago Manual of Style*, 15th Edition

- Norman Walsh's DocBook: The Definitive Guide (TDG)

  `http://www.docbook.org/tdg/en/html/docbook.html`

1 Creating a Consistent Structure

It is important to maintain a relatively consistent structure within a document. This structure may vary for different books or projects. For consistency, each project should have its own guidelines offered here, in a separate project document, or through templates.

1.1 Lessons for Lizards

The standard base structure of LfL is reflected in its template. Find the template in LfL's SVN. The project is still in development stages. Once the project is more stable, guidelines will be added here.

1.2 Official SUSE Manuals

A SUSE manual consists of the following mandatory elements: title page, copyright notice, table of contents, preface, and chapters split in sections. Optionally, use parts, appendixes, a glossary, and an index, if appropriate. Find descriptions and usage of these elements in the following list.

Title Page
: The title page contains logo, title, and subtitle.

Imprint
: The second page (imprint) lists edition, copyright notice, and the names of all contributors to the manual. Enter the names in the following files:

`authors-chapters.xml`
: List authors.

`authors-trans.xml`
: List all translators who worked on the manual.

`authors-office.xml`
: List the members of the documentation department who wrote and edited the manual. This file is only used in internally-produced documentation.

Table of Contents
: The table of contents is generated automatically.

Preface
: The preface of a book contains a brief overview of the scope and content of the manual, information about how to read it, and typographical conventions.

Parts
: Parts structure a manual with many chapters to improve the usability of a book. Use nouns as part titles and keep them clear and concise. Some typical part titles would be “Installation” or “Network”.

Chapters
: Chapters typically consist of the following elements (appendixes may be regarded an exception):

Figure 1 *Structure of a Chapter in SUSE Documentation (* for optional element)*

```
chapter
|
+--- title
|
+--- abstract
|
+--- section
.    |
.    +--- introductory text
.    |
.    +--- procedure environment*
.    |
.    +--- subsection
.    .
.    .
.    .
.    +--- subsection
.    |
.    +--- troubleshooting*
.    |
.    +--- for more information*
.
+--- troubleshooting*
|
+--- for more information*
```

Abstract

The abstract should provide a brief summary of the information provided in the chapter. Rather than writing about what the chapter does, summarize the information the chapter contains. This gives the reader a clear idea of what he or she can learn by reading the chapter. Example 1, “An Example Abstract” (page 6) is an example abstract from the chapter about file systems.

Example 1 *An Example Abstract*

openSUSE 10.2 ships with a number of different file systems, including ReiserFS, Ext2, Ext3, and XFS, from which to choose at installation time. Each file system has its own advantages and disadvantages that can make it more suited to a scenario. Professional high-performance setups may require a different choice of file system than a home user's setup.

Introductions

Any introductory information follows directly after the abstract and should not be placed in a separate section.

Sections

Structure the detailed information in sections for better user orientation and to allow the experienced user to find the desired aspect of the topic easily.

Create sections in analogy to the main tasks, like installing, configuring, monitoring, and administering. If helpful, split sections into subsections. Sections start with an introductory paragraph outlining the focus of the section. If the section works with a sequential task, you can follow this introduction with a procedure description, as discussed in Section 3, “Creating Procedures” (page 21), which clearly states all the necessary steps to accomplish a task. Experienced users familiar with the topic can follow this brief sketch. For less experienced readers, each step of the procedure can contain a link to a subsection where the necessary background is provided and the action explained in more detail. In HTML and PDF online documentation, these links are clickable, providing easy access to the additional information.

Troubleshooting Section

If you know of common mistakes, pitfalls, or problems a user might encounter in this situation, list them in this section. Use the title “Troubleshooting” for all these types of sections in all manuals. Structure the section in a symptom and solution pattern.

For More Information

As described in Section 2.2, “Addresses” (page 8), this section contains links to all sources of information that might prove helpful in the given context. Follow the general referencing guidelines in Section 2.7, “Cross-References and References to External Resources” (page 11) when creating this kind of section.

Glossary

The optional glossary contains important terms in alphabetical order and provides a brief explanation.

2 Writing and Editing

These rules are intentionally kept brief and simple. They contain specifics related to SUSE and openSUSE documentation and sometimes standard English rules. Where relevant, specific markup may be mentioned to assist in maintaining consistency in SUSE documentation.

2.1 Abbreviations and Acronyms

Most abbreviations may not be used. “etc.” may be used in parentheses and some rare instances, but not in combination with phrases like “for example” or “such as.” It should be followed by a comma.

Abbreviations of common units of measure may also be used. Do not create a plural of these abbreviations by adding an “s.” Use a nonbreaking (`&nbsp;`) space between the numeral and multiple-letter abbreviations. Do not use a space before single-letter abbreviations.

A header must not contain both the acronym and the meaning. Common acronyms can be used in the header and written out in the following text.

Once the acronym has been presented in the written out form, use the acronym version in the rest of a chapter. Because sometimes chapters and parts are used in different documents, it is important to map acronym and meaning at least once per chapter. Be consistent with this mapping. If you use the acronym followed by the written out form in one chapter, try to do this in all others.

Form plural forms of acronyms by adding a lowercase “s,” for example, “CDs” and “BIOSs.” Because some rare acronyms create plurals by adding “'s,” avoid using possessive forms of acronyms for reasons of clarity.

2.2 Addresses

SUSE documentation contains “For More Information” sections at the end of most chapters to point to further sources of information regarding the topic. Whenever possible, links and addresses are grouped there and presented by means of a `variablelist` environment (link plus explanatory remarks). Do not use a list environment for only one link.

If you need to present a large number of links, group them by topic and create a separate list environment for each group. Provide a comprehensive title for each of the groups or an introductory sentence.

Translators are normally provided with localized versions of links in the comments of the source code. If none is provided, the English default is used as fallback.

You can provide links or addresses directly in the main sections, but are encouraged to use and refer to the “For More Information” section wherever possible. Both automated checks and the use of editor comments for translators ensure that the links are kept up-to-date.

For detailed information about external and internal references and addresses in SUSE documents, refer to Section 2.7, “Cross-References and References to External Resources” (page 11).

2.3 Admonitory and Advisory Paragraphs

SUSE uses four different kinds of admonitory and advisory paragraphs. Because these elements are formatted in a way to draw the reader's attention to them, their use should be limited to specific situations. Additionally, avoid using more than one per page.

`warning`

These elements warn of security issues, potential loss of data, damage to equipment (hardware), or physical hazards to the user. A warning must always precede the action to which it applies. A typical example of a warning used in SUSE documentation is:

Example 2 *A Typical Example of a SUSE Warning*

```
<warning>
 <title>Creating File Systems</title>
  <para>
    Formatting may take up to several hours depending
    on the partition size. Do not try to interrupt
    this process. This could result in a corrupted file
    system that renders the installation
    unusable. Proceed with the installation after
    formatting has been completed.
  </para>
</warning>
```

A warning consists of the warning headline, a title stating the action considered dangerous, followed by a descriptive paragraph explaining the nature and origin of the danger. Clearly state what could happen if the reader ignored the warning. Finally, tell the reader how to avoid the dangerous situation.

`important`
: These elements introduce vital information that deserves particular attention. Similar to the warning element, `important` includes a title stating the topic covered in the following explanation.

`tip`
: A `tip` is used to introduce guidelines or tips that are helpful to the user. Using an appropriate title is strongly recommended.

`note`
: The `note` element is useful for emphasizing information that differs from previous versions, providing important information that does not fit into the paragraph, and pointing out architecture-specific information. Using an appropriate title is strongly recommended.

2.4 Audience

It is generally not necessary to describe the intended audience in the text, but keep it in mind when writing the document. Adjust the tone, style, and technical level of the text based on the intended audience. Do not insult them by saying something is obvious.

Aim the document at the reader by describing tasks the reader may need to complete rather than the organization of the program. For example, when describing YaST, focus on how to configure the devices and services the user needs rather than the module structure and available dialogs.

Try to focus on the positives: what the program can do rather than its limitations. Be honest in your documentation and avoid absolutes and the exaggerations of marketing-speak.

2.5 Capitalization

Use title-style capitalization for all types of headings and titles, including figure and table titles. This style is explained in *Chicago* 8.167.

This style can be very complicated to implement, but the following these simple rules is a good start:

1. Capitalize the first word no matter what it is

2. Capitalize all major words

3. Lowercase *the*, *an*, *a*, *and*, *but*, *for*, *or*, *nor*, *to*, and *as*

4. Lowercase prepositions, such as *up*, *in*, *down*, and *of*

5. Capitalize the last word

In variable lists, capitalize the term in title-style unless it is a complete sentence. In all list items, use sentence-style capitalization (capitalize the first letter and proper nouns only). Use title-style for row and column labels in tables.

2.6 Contractions

SUSE and openSUSE documentation does not normally permit the use of contractions. Contractions are only permitted in casual documents and texts. A good example of appropriate use of contractions is the SuSE tour that was included in older releases. It provided a friendly and somewhat humorous introduction to applications and was aimed predominantly at the the home user.

2.7 Cross-References and References to External Resources

Use `xref` when you refer to an appendix, chapter, part, preface, section, table, figure, or example. The referenced element needs an `id` attribute. When referencing these elements, do not insert a text label like “appendix”, “chapter”, “part”, “preface”, “section”, “table”, “figure”, and “example”. These are inserted automatically.

Do not insert a reference to a paragraph (`para`). If you need a reference to a list, insert `id` and `xreflabel`.

References to man pages and info pages should generally be made in the format “the man page of `command`” or “the info page of `command`”. In a situation where the category of the page is needed, add “(`man 9 command`)” to the end.

Use ulink for a URL, like http://www.suse.de. Always use the correct protocol prefix (http, ftp, mailto) to enable the automated link check. Never use filename for this kind of link. Anything that should show up as a clickable link in the final HTML and PDF docs must bear the correct markup. Otherwise risk breaking the link checking tools and rob the user of extra functionality, such as click to reach extra external documents. Do not enter a text label for the ulink so that the URL itself appears in printed versions.

Do not use ulink and the file: prefix to reference filenames. Use filename for files.

Do not use links across documents where ID and IDREF linking is not possible and ulink is inappropriate. For references to information in SUSE books outside the current document set, use the appropriate macro for the book title. The reference should not refer to a specific chapter or section number or title because these things change too easily. Instead, make a reference using the document title but no specific headers.

Insert references to external (not SUSE or openSUSE) books in the format "Title by Author (ISBN #)." Inclusion of the ISBN is optional. Place the title in citetitle tags. For example:

```
<citetitle>Indexing Books</citetitle> by Nancy C. Mulvany
(ISBN 0-226-55014-1) is a useful guide for indexing.
```

It is useful to provide language-specific references in the translated books. If the authors know of language-specific versions, they can include them in comments to the translators. As time permits, translators should look for corresponding language-specific resources. For URLs, provide only the language-specific version of a site. Use the English as a fallback. For books, provide the title of the translated version along with the original title if such a translation exists.

2.8 Discriminatory Language

Refer to *Chicago* 5.43, 5.51, and 5.78 for suggestions of how to avoid gender bias. Basically, rewrite the sentence to avoid indications of gender. If possible, use plural to allow use of “they” as the pronoun. If these attempts fail, use “he or she.”

For example users, use the various mascots, like Tux, Geeko, and Wilber.

2.9 Illustrative Block Elements

Graphics and Figures

Screen shots and graphics should only be used when helpful for following the explanation. Using too many screen shots can confuse the reader even more. If you intend to use screen shots in a document, consider the following issues first:

- A screen shot should illustrate a complex situation where the user cannot easily follow the text otherwise.

- Do not use screen shots of confirmation pop-ups, progress bars, and other widgets that do not require user action and do not provide crucial information.

- Do not illustrate every step of a procedure with a screen shot. In most cases, selecting a few screen shots to show the most important dialogs or choices during the whole process is sufficient.

If you intend to use graphics in documentation, think of the following issues first:

- KISS (keep it short and simple). Custom graphics should always simplify complex issues and not confuse the reader by their complexity. Embedding too much text in the graphic adds to the complexity. Use as little text as possible and be aware of localization issues, such as differing lengths of strings in different languages.

- Provide an appropriate reference in the text. The layout process should make an effort to ensure that figures are not placed too far from the related text.

Callouts

Callout numbers are only printed if there are 10 or fewer per example. Using more than 10 would be distracting.

__Example 3__ A Configuration File with Callouts

```
color white/blue black/light-gray   ❶
default 0   ❷
timeout 8   ❸
gfxmenu (hd0,0)/boot/message

title SUSE LINUX 9.1.42
    kernel (hd0,0)/boot/vmlinuz root=/dev/hda1 vga=0x31a selinux=0
    splash=silent resume=/dev/hda2 desktop elevator=as  showopts
    initrd (hd0,0)/boot/initrd

title Disk
    root (fd0)
    chainloader +1
```

❶ Here, define the colors used by the GRUB menu.

❷ Define the default option.

❸ Select a time-out after which booting starts automatically.

For details on the markup of this element, refer to Section A.3.3, “Callouts” (page 46).

Examples

Like figures and screen shots, examples are used to illustrate complex processes. Most of the rules established for the use of these elements in Section “Graphics and Figures” (page 13) also apply to examples.

If you intend to use examples, consider the following:

- Examples are used to present long commands and their output, lines of code, configuration files, excerpts of configuration files, and output, such as system messages.

- Make sure that the example is crucial to the understanding of the process as a whole. Like figures and screen shots, examples stick out from the text and are distracting when overused or used improperly.

- If you need to present just a function or application call on the command line, it is normally sufficient to do so inline. However, if you need to feature both a command and its output on screen, use an example environment.

Example 4 *Calling ps to Get Process Information*

```
ps -xa

5170 ?        S      0:00 kdeinit: khotkeys
5172 ?        S      0:02 kdeinit: kdesktop
5174 ?        S      0:04 kdeinit: kicker
5177 ?        S      0:00 kdeinit: kpowersave
5179 ?        S      0:00 kdeinit: klipper
```

- The example environment should contain both a title and an ID to enable referencing and easier coordination with the text.

For details on markup, refer to Section A.3.2, “Examples” (page 45).

2.10 Headings

Each level of heading should have two or more headings. Lonely subheadings are not allowed. Make headings in a group parallel by using a similar structure for the wording.

Keep headings short and simple. Do not use both an acronym and the written-out form in a heading.

Provide content between a heading and its subheadings. In some cases, this is only a small paragraph of introductory information. In other cases, more information can be provided.

2.11 Lists

SUSE and openSUSE documentation uses the following types of lists (the respective XML tags are given in parentheses):

Bullet Lists (`itemizedlist`)

> See `http://www.docbook.org/tdg/en/html/itemizedlist.html` for details of markup.

Numbered Lists (`orderedlist`)

> See `http://www.docbook.org/tdg/en/html/orderedlist.html` for details of markup.

Descriptive Lists (`variablelist`)

See `http://www.docbook.org/tdg/en/html/variablelist.html` for details on markup.

Consider the following general rules when creating or editing list environments:

- List environments should be used cautiously. Their markup is quite distinct and overusing them might disrupt the text flow.
- Always introduce a list in the text. If needed for reference or better coordination with the related text, add an optional title and an `id` attribute.
- A list must contain at least two items. If items are short and simple in structure, consider incorporating them in the flowing text instead of creating a list environment.
- Use sentence-style capitalization for the list items. Use title-style capitalization for terms in descriptive lists.
- If using multiple sentences as a list item, end all items in that list with a period.
- Make sure that the items are grammatically parallel constructions providing a pattern that makes it easier to follow the text.

The different types can be nested, but avoid nesting too many levels. When nesting different types of lists or introducing multiple levels of one type, always consider the flow of the text. It is rarely necessary to have more than one level in a list. Using too many levels disrupts the text flow, confuses readers, and heavily influences the document layout. Consider these issues carefully before nesting lists or introducing another list level. Restructuring the information using a well-balanced combination of lists and flowing texts may prove a good approach.

Bullet Lists

Bullet lists are often used to provide an overview of information or to introduce or summarize information. They should be used when the order of list items is irrelevant.

If needed, assign an `id` attribute to an itemized list and add a title to reference it. Individual list items may be referenced by assigning an `id` and an `xreflabel` attribute. The entry is then identified by the value of `id` and the value of `xreflabel` is printed as the label of this item. See Section 2.7, “Cross-References and References to External

Resources” (page 11) and Section A.4.6, “Referencing” (page 51) for detailed information.

The following example of an itemized list is from a “For More Information” sections.

Example 5 *A Bullet List*

Linux Information

- `http://www.suse.com`
- `http://www.novell.com/linux`

Numbered Lists

Use numbered lists when items have a strict order, hierarchy, or importance. If order is not relevant, use a bullet list or a descriptive list. Do not use numbered lists to describe procedures. Complex sequential actions are better described by means of the procedure environment (see Section 3, “Creating Procedures” (page 21) for a detailed description).

If needed, assign an `id` attribute to a numbered list and add a title to it in order to reference it. Individual list items may be referenced by assigning an `id` and `xreflabel` attribute. The entry is then identified by the value of `id` and the value of `xreflabel` is printed as the label of this item. See Section 2.7, “Cross-References and References to External Resources” (page 11) and Section A.4.6, “Referencing” (page 51) for detailed information.

Example 6 *A Numbered List*

1. The machine's network is configured properly.
2. The machine's exact system time is maintained by synchronizing with a time server. This is necessary because parts of the HTTP protocol depend on the correct time.
3. The latest security updates are installed. If in doubt, run an online update.
4. The default Web server port (port 80) is opened in the firewall. For this, configure the SUSEFirewall2 to allow the service *HTTP Server* in the external zone. This can be done using YaST.

Descriptive Lists

Use descriptive lists when defining terms or describing options. Each item of a descriptive list contains a short term that is then further explained by means of an explanatory paragraph.

Use title-style capitalization for the term. Use sentence-style for the list item.

You may assign an `id` attribute to the list and add a title to it in order to reference it. Individual list items may be referenced by assigning an `id`. The entry is then identified by the value of `id` and referenced by the term.

Example 7 *A Descriptive List*

This book consists of several parts:

Installation
: Learn about the installation and initial configuration of a Linux system.

System
: Get a basic understanding of the system components.

2.12 Numbers and Measurements

Follow *Chicago* 9.6 for numbers. This is basically to write out single-digit numbers (0–9) and use numerals for all others. For measurements, always use numerals when an abbreviation is used for the unit.

2.13 Possessives

Do not use possessives of acronyms and trademarked terms. Unless use of a possessive with an inanimate object improves the clarity of a sentence, avoid inanimate possessives.

2.14 Punctuation

Use commas between all items in a series. For example, write “a, b, and c.” Use commas around phrases like “for example” and “that is.” Introductory phrases in the beginning of a sentence are normally followed by a comma. For example, “Before using YaST Online Update, configure a network connection.”

Semicolons (;) are mainly used to join two independent but related sentences. They can also be used in place of commas in a complicated series. It is preferable to avoid them when possible and instead break the text into smaller pieces.

em dashes (—) are punctuation dashes. Use them sparingly and do not surround them with spaces. en dashes (−) should be used between numbers in a range in tables and figures. Also use them for a hyphenated term when a two-word term is part of the hyphenated term (file system–specific). Additionally, use them as the symbols for minus signs.

Do not use slashes except when they are part of a standard technical term, such as TCP/IP or client/server. Avoid the use of exclamation marks (!).

2.15 Sentence Structure

Keep sentences clear and direct. Avoid complicated clauses and try to keep sentences short (under 20 words). Avoid joining sentences with semicolons (;) if they can be as easily understood as two independent sentences. Make sure that the relationship between subject, verb, and object are clear. Avoid ending a sentence with a preposition where reasonable.

2.16 Tables

Use tables to present a large number of similar facts. A table makes them easy to scan and compare when presenting the same information textually would require several paragraphs.

Always keep tables as simple as possible. They should not require extensive explanations in the related text and should make sense even to those readers who are not very familiar with the topic.

A table always has a title. If you assign an `id`, it can be identified and referenced by this attribute.

Value and description pairs are better handled by means of a descriptive list.

Example 8 *A Typical Table in SUSE Documentation*

File System	File Size (Bytes)	File System Size (Bytes)
Ext2 or Ext3 (1 kB block size)	2^{34} (16 GB)	2^{41} (2 TB)
Ext2 or Ext3 (2 kB block size)	2^{38} (256 GB)	2^{43} (8 TB)

2.17 Tone, Mood, Voice

Maintain a professional tone. Except in casual documents, do not use contractions. Do not use humor. It does not translate well and may be misunderstood by nonnative speakers.

Use second person (you) to refer to the reader. Normally the reader is the user or administrator that does the actions described. Do not overuse “you.” It is often understood, especially in directions.

Use active voice. Active voice means that the subject of the sentence performs the verb. For example, “The root user performs administrative tasks” is active voice. “Administrative tasks are performed by the root user” is passive voice. Passive voice can be used occasionally if there is a need to place emphasis on the object of the verb or if the doer of the action is unknown. “A SAMBA server must be configured in the network” is an example of proper use of passive voice. The emphasis is on the server, not on the person configuring it.

2.18 Verbs

Use the simple present tense in almost all cases. This also applies for sentences with “if” or “when” clauses. “If you do this, something happens.” “When you do that, something else happens.”

3 Creating Procedures

Procedures are used to describe a task. A procedure consists of several steps that are usually performed sequentially. A procedure environment consists of the following elements:

- An optional title. If you need to reference the procedure as a whole, set an `id`. The title is then used as the label of the reference.

- An introductory phrase establishing the purpose of the procedure. If necessary, add the preconditions of the operation.

- Steps and substeps describing what the reader must do. If needed, add an explanatory paragraph, but keep it short. Do not use only one substep in a step or one step in a procedure.

- Alternative actions are introduced by a substep environment containing two steps linked by “or.” Each optional step must have a `performance` attribute set to `optional`.

Procedure 1 *Adding Users with YaST*

To add new users to a Linux system, perform the following steps:

1 Start the YaST module (*YaST* > *User Management*)

2 Set the filter (type of users).

3 Add the first user:

  3a Click *Add* to enter the *Add a New User* dialog.

  3b Enter the user data and click *Create*.

  Repeat this procedure for each user to add.

4 After the last user has been added, leave the YaST module by clicking *Finish*.

If you need to structure complex tasks, procedure environments are a means to provide an overview of the whole task and its sequential character. Introduce the steps, substeps, and alternatives by a procedure environment. A step may contain a link to an explanatory subsection providing further details on the step.

4 Presenting Computer Elements

When documenting software, it is frequently necessary to describe the software and how to interact with it. The way to do this has been standardized for consistency.

Only specify the software component in ambiguous context. Only put literal user input or output on a separate line when it is too long to be included in the flowing text or is a line that must be added to a file.

4.1 Command Presentation

When referring to an application, do not use any markup at all:

OpenOffice.org is a powerful open source office suite.

Linux commands are case-sensitive, so always present them as they appear in the system. Commands can be embedded in the flowing text or presented as part of screen environments or examples. In flowing text, use `command` when referring to an actual command you would use on a command line:

Use `OOo` to start OpenOffice.org from the command line.

Place options inside the command tags. Options can also be used in text, separately from `command`. See Section A.4.3, “Commands and Options” (page 50).

Do not use any markup for commands as part of a `screen` environment. Anything inside this environment appears verbatim.

4.2 Variables

There are currently two different kinds of variable environments used in SUSE documentation:

- Substitutable contents that the user should replace with appropriate values. For example:

  To list the contents of a directory, execute `ls` *`dir`*.

- References to names of variables rather than to their value. Example:

  To select another display manager, start the YaST
  sysconfig editor and change the value of
  `DISPLAYMANAGER`.

4.3 Describing Graphical User Interfaces

Keys

SUSE uses the standard elements and transformations offered by DocBook V. 4.3 and the current XSLT stylesheets by Norman Walsh. For details on key markup, refer to Section A.4.1, “Markup of Keys” (page 48).

Always use the capitalized version of characters, because this is how they are printed on the keyboard. If you need to refer to a capitalized character, use Shift + Z, for example. Introduce this convention by means of the “Typographical Conventions” section of the introduction.

With DocBook V. 4.3, the localization of the identifiers for both modifiers and special keys is handled by DocBook itself. Just use the `keycap` tag and an appropriate `function` value. Find details in Section A.4.1, “Markup of Keys” (page 48).

GUI Elements

When using labels of GUI elements, do not include ending punctuation like `...` or `:`. Whenever possible, refer to the elements without identifying them as a special type of dialog element. For example, use “Click *OK*” rather than “Click the *OK* button.” However, some situations require a more specific wording, for example, if a dialog has many different graphical elements.

For detailed instructions for the appropriate markup, especially concerning buttons, labels, and menu structures, refer to Section A.4.2, “GUI Menus” (page 49).

4.4 Naming Conventions

SUSE documentation is limited to describing the Linux operating system and open source applications. The following sections summarize the common naming and markup principles that apply to SUSE and openSUSE documentation.

Directories and Filenames

Linux does not know any drive names. Under Linux, everything is considered a file (even a directory is considered a special kind of file). Therefore, the naming and markup conventions are the same for “drives” (for example, hard drives, CD-ROM drives, or floppy drives), directories, or files.

The layout for filenames and directory names is the same. See the following example:

Open the file `suseconfig` under `/etc`. Access an USB
flash drive via `/media/usb-xxx-yz`.

Use slashes properly when dealing with file and directory names:

- Use a leading slash to indicate that the directory is reached from the root of the file system. If the directory lies below the current working directory, do not use a leading slash.

- Do not use a trailing slash to distinguish between a file and a directory. When differentiation is needed, provide it textually.

Linux is a case-sensitive operating system, so use capitals exactly as they appear in the file system. For details on markup, refer to Section A.4.5, “Directories and Filenames” (page 51).

Hostnames and IP Addresses

Writers should use a well-defined set of IP addresses, hostnames, and usernames for their network documentation. The values are provided by means of entities (see Section 5, “Entities” (page 25)).

IP Addresses
: Currently, we are using only a class C net (netmask 255.255.255.0) for our examples. To create examples using a larger setup, select addresses from the private network ranges.

Hostnames and Domains
: SUSE uses `www.example.com` as a domain for most examples, because this is a real domain that was designed for use in documentation. Hostnames are taken from the solar system (such as `sun` or `earth`).

Usernames
: SUSE currently uses mascots like `geeko`, `tux`, and `wilber` as example users.

5 Entities

Entities are a kind of “macro” in XML, but only with text expansion capabilities. They are used for several purposes:

- Representation of characters that cannot be displayed or entered otherwise.

- Integration of external files by entities representing references to their filenames.

- Text expansion to any content that is needed.

Once an entity is defined, it can be used everywhere. Entities reduce the risk of translation errors and increase consistency because they are translated once and automatically expanded elsewhere.

5.1 General Notes on the Handling of Entities

SUSE uses a small set of custom entities. This set is localized for each supported language. Find the SUSE set of entities in the file `entity-decl.ent` under each documentation project and each language. When the set has been modified for a new product release, this file must be updated in the supported languages as well.

Each header of a SUSE XML file includes the entity declaration file (by means of an entity):

```
<!ENTITY % entities SYSTEM "../entity-decl.ent">
%entities;
```

Example 9 *Excerpt from a SUSE entity-decl.ent*

```
<!ENTITY❶ exampleserver❷   "sun❸">
<!ENTITY exampleserverip    "&exampledomainip;.20">
<!ENTITY exampleserverfq    "&exampleserver;.&exampledomain;"❹>
```

❶ Tells DocBook which kind of declaration to expect.

❷ Defines the entity name.

❸ Sets the value to which the processed entity should be expanded.

❹ Nests an entity in the value.

A few hints for working with entities in SUSE documentation projects:

- If there should be any need for defining special entities (for example, for localization purposes), add the new definitions to the appropriate `entity-decl.ent` file. Never include these definitions in the file header. Always check with the documentation team members before introducing new entities.

- When translating entities, translate the value, but never change the entity name (see Example 9, “Excerpt from a SUSE entity-decl.ent” (page 26)).

- Always use the exact notation of the entities when translating files to avoid processing errors.

- Make sure that you use proper UTF-8 encoding when editing and saving the entity declaration file or any of the SUSE XML files.

5.2 Entities Used in Documentation Projects

The following provides some background on the most important entities used in creating documentation for SUSE products. To be able to adjust product names of platform branding even past any deadlines, a set of entities must represent the frequently-changing product names. The same applies to book titles. An `entity-decl.ent` file contains several different categories of entities. These are:

General Entities
: These feature mostly network IP addresses, hostnames, and usernames.

Books
: If possible, any book created by our team has a title entity in case sudden changes are necessary.

Platforms
: To avoid touching our content whenever a hardware vendor rebrands its products, we use entities for all hardware architectures referenced in the books.

Products
: We maintain a list of entities of all SUSE-based product names and some other products and applications for which we write documentation.

There are several guidelines to consider when working with product entities for SUSE documentation:

Entity Selection
: SUSE uses `&product;` to identify the product for which the documentation is built (the text applies to any product). The value of this entity is set once per release and expands to the name of the current product.

Example 10 *Proper use of the &product; Entity*

`&product;` includes openLDAP.

If you need to reference a particular product, use the specific entity (plus version if necessary).

Example 11 *Context-Based Use of Entities*

The ext2 file system has been included in the `&suse;` kernel since `&suselinux;` XYZ.

Trademarks

Trademarks and openSUSE

At this point, only internally-produced documentation is required to follow the trademark guidelines. These rules can be skipped for documentation produced by the openSUSE community.

Any Novell product referenced in our documentation is trademarked one way or the other. Currently we have entities for SLES, SLED, openSUSE, and the obsolete SUSE Linux stuff. Adhere to the following rules when dealing with these terms:

- Never use trademarks in headings.
- Only use the trademarked version on the first occurrence of the product's name in a chapter.

Product entities carrying any sort of trademark end in `&reg;`.

Acronyms

Acronyms and openSUSE

At this point, only internally-produced documentation is required to follow the acronym guidelines. These rules can be skipped for documentation produced by the openSUSE community.

Avoid using acronyms for products because they are not the proper product names. However, there might be occasions where you just need to have an acronym. In these cases, use the acronym entity that always ends in `a;`, such as `&sleda;` or `&slesa;`.

6 Indexing

An index is one of the most important parts of a book. Index terms should be used to provide the reader with access to useful information. Insert the term as close to the relevant text as possible to ensure that the page reference points to the correct location. When more than a few paragraphs contain information that should be indexed under this term, use a page range. A user referring to the index can then get a quick overview of how much information is provided at the different locations and have a better chance of finding needed information quickly. Up to three levels of index items can be used.

Only index text that really provides useful information to the reader. For example, a section about configuring an application should be indexed to the application, but a passing reference to the application should not be indexed at all. Try to think like a potential user of the manual. Is a user likely to be looking for that information? Some examples of things that should not be indexed include “For More Information” sections, things mentioned in the abstract of a chapter (index where it is actually discussed, not where it is mentioned in the abstract), and a command used for some other purpose (do not index `cd` where it is used to change directories as part of a procedure, but do index a section discussing appropriate usage of the command itself).

Select the terms to use carefully because consistency is important. Write nouns in the plural and verbs in a gerund form (-ing). Capitalize terms only if they would be capitalized if used in the middle of a sentence.

When selecting terms, try to consider what else is included in the manual. If the manual only includes a small amount of information for the topic, general index items are probably more appropriate. If indexing a chapter about an application, much more specific index terms are needed so the user can find exactly the information he or she needs. Also consider that the user might not know what something is called. It is not helpful only to index Konqueror under its name if the user might want a Web browser and not know what applications function as a Web browser. In a case like that, you may need to index general information both places.

If you have time when working on indexing, check the built index to see how well your terms integrate with existing items. Try to make your terms consistent with other items used. A final index “polishing” of an entire index is part of the editing and layout process. A few tips for evaluating and adjusting the index of an entire document:

- If a term has six or more page references, check those references and see if it is possible to create subitems.

- If a term has only a single subitem (or a subitem has only a single sub-subitem), consider deleting the subitem.

- Check for spelling errors or inconsistencies that result in multiple items in the index.

- If there are copious third-level entries or third-level entries with too many references, consider adjusting the index structure to use a more specific primary term.

- Check all *see* and *see also* references for consistency. An item should not have both a page and a see reference. Page references and *see also* together is acceptable but should be used sparingly.

Example 12 *A Plain Index Entry*

```
<para>
  To configure DNS<indexterm>
    <primary>DNS</primary>
    <secondary>configuring</secondary>
  </indexterm> use ...
</para>
```

Example 13 *An Index Range*

```
<indexterm class="startofrange" id="idx.Installation">
  <primary>installing</primary>
    </indexterm>
...
<indexterm class="endofrange" startref="idx.Installation"/>
```

Example 14 *A See Index Entry*

```
<indexterm>
  <primary>installing</primary>
    <secondary>boot loaders</secondary>
     <see>GRUB</see>
</indexterm>
```

7 Preparing Glossary Entries

An optional glossary contains terms and their definitions. Make sure that the glossary entries are appropriate to the intended audience. Define unfamiliar terms and specialized jargon.

Define infinitive forms of verbs and singular nouns. Use lowercase for the term unless it is a proper noun.

Use cross-references to link acronyms with their written out forms. Define the written out form. Use a *See* reference for the acronym form to link it to the defined written out form.

The markup for a glossary entry is shown in Example 15, “A Typical Example of a Glossary” (page 31).

Example 15 *A Typical Example of a Glossary*

```
<glossary>
 <title>Glossary</title>
 <glossdiv>
  <title>D</title>
  <glossentry>
   <glossterm>DocBook</glossterm>
   <glossdef>
    <para>
     DocBook provides a system for writing structured documents
     using SGML or XML. It is particularly well-suited to books
     and papers about computer hardware and software, though it
     is by no means limited to them.
    </para>
   </glossdef>
  </glossentry>
 </glossdiv>
</glossary>
```

8 Spelling Terms

Because multiple spellings of a variety of words and terms are in common usage, we have standardized on a variant to use in our documentation. Terms in this table are capitalized as they would be in the middle of a sentence. Some words listed here are provided because they are often written incorrectly by nonnative speakers. For words not listed here, *Merriam-Webster's Collegiate Dictionary, Eleventh Edition* is considered authoritative.

Table 1 Terms Common in SUSE Documentation

Word or Phrase	Usage Guideline
3D	no hyphen, capitalize D
32-bit	hyphened, lowercase B
address book	two words
back-end	hyphenate
back up	verb
backup	noun
Bash	capitalize according to the GNU Bash manual
boot disk	disk for starting the system
boot loader	two words
cannot	one word
case-sensitive, case-insensitive	always hyphenate
certificate authority	acronym is CA; not “certification authority”
check box	two words
checklist	one word
check mark	two words
chipset	one word
client/server	a slash is accepted here
command line	never hyphenated

Word or Phrase	Usage Guideline
control center	a generic term as in the YaST control center or the KDE control center
deselect	verb
DHCP	capitalized
dial-up	use only as an adjective
DNS	do not use DNS name server; capitalize the acronym, but lowercase the expansion
double-click	hyphenate, do not use with “on”
e-mail	always hyphenated
Ethernet	capitalized
facilitate	avoid using
filename	one word
file server	two words
file system	two words
framebuffer	one word
front-end	hyphenate
The GIMP	project standard for application
GNOME	project standard for the desktop
GRUB	capitalize
GUI	capitalized

Word or Phrase	Usage Guideline
hard disk	two words
hard link	two words
have to	use "need to" or "must" instead
home page	two words
hostname	one word
hotplug, hotplugging, hotpluggable	note spellings
hypervisor	one word, lowercase
infrared	one word, no hyphen
installation source	correct term for valid sources of installation data for SUSE Linux and openSUSE; when reasonable, mention that online repositories that are valid installation sources are available
Internet	capitalize first letter
intranet	lowercase
I/O port	slash in I/O
IPsec	note capitalization
journaling	one L
Kdump	capitalize if application
kdump	lowercase if command
Kernel	capitalize

Word or Phrase	Usage Guideline
Kprobes	capitalize if application
kprobes	lowercase if command
left-click	hyphenate, do not follow with “on”
local host, localhost	normally two words, one word for default name of the local host
log file	two words
log in, log out, login	two words for verbs, one for noun; do not use “log on” or “log off”
loopback device	loopback is one word and not hyphenated
mail server	two words
Maildir	note capitalization; this is a specific format, not a generic reference to the directory in which mail is stored
mainboard	one word
man page	two words
Mbox	note capitalization
metadata	one word
middle-click	hyphenate, do not follow with "on"
mount point	two words
mouse button	a mouse has buttons, not keys
multitasking	one word, no hyphen

Word or Phrase	**Usage Guideline**
multiuser	do not hyphenate
name server	two words
NFS	an NFS server, an NFS client, etc.
NIS	a NIS client, a NIS server, etc.
paravirtualized	one word
pathname	one word. can often be replaced with “path”
plug-in	hyphenate noun or adjective
pop-up	hyphenate
port	on port (not “at port”)
PostScript	use this capitalization
print queue	print not printer
print spooler	print not printer
proxy	use as a noun only
RAM disk	two words, note capitalization
README	Use all caps for general references because this is the most common form in Linux; use file's capitalization for specific references
read-only	hyphenate
reconfigure	one word, no hyphen
re-create	hyphenate

Word or Phrase	**Usage Guideline**
right-click	hyphenate, do not follow with “on”
RPM	capitalized
runlevel	one word
runtime	one word, no hyphen
screen shot	two words
screen saver	two words
SCSI	an SCSI device
set up, setup	two words as verb; one word as adjective or noun
slider	not slidebar or other variations
spec file	two words
SSH	capitalized
stand-alone	hyphenate
start-up	hyphenate
status bar	two words
tarball	one word
taskbar	one word
TFTP	capitalized
time stamp	two words
title bar	two words

Word or Phrase	Usage Guideline
toolbar	one word
tool tip	two words
uninstall	no hyphen, never de-install
unselected	the state of not being selected
username	one word
utilize	use “use”
virtualization	lowercase
VLAN	capitalize
Web	capitalize first letter
Web page	two words
Web server	two words
Web site	two words
whether	do not add “or not”
wild card	two words
X Window System	do not shorten to X Windows, note capitalization
Xen	use this capitalization
Xend	capitalized
Zypper	capitalize if application
zypper	lowercase if command

9 Inserting Comments

Use these standard comment and remark formats to simplify editing and translation.

9.1 Remarks

Remarks are used for the majority of editorial comments. It is possible to have them be visible in the built document, improving the editing process. Remarks must be placed before or after a para but inside section tags.

Include a `role` to show the importance and type of remark. Available roles and their uses are:

xml
: This type of remark is used for validation issues and questions and comments regarding markup (wrong tag, validation problems the commenter cannot resolve without help, etc.).

layout
: Use this for remarks related to the layout. For example, use it for comments suggesting a change in list type or those related to line break problems in the output.

structural
: Structural remarks relate to how the text is put together. It can be about content that should be placed elsewhere or the level of headings.

clarity
: This role should be used when text is unclear or changes are proposed to improve the wording.

needinfo
: The author should use this role to report when he or she is waiting for information from a developer or other person or to request information.

trans
: Use this for comments to translators.

generic
: Use this role for all remarks that do not fit into one of the other categories.

The writer of the remark should start the remark with his or her username followed by a colon (:). Others responding to the remark should insert their username, a colon, and comments inside the tags but starting on a new line. Comments should be deleted by the original writer when they consider the issue resolved. The following is an example remark:

```
<remark role="needinfo">tux: couldn't find the option for foo
geeko: see /usr/share/doc/foo.html</remark>
```

9.2 XML Comments

XML comments are primarily used for temporarily disabling portions of text. Because they can appear at any point in a file, they can also be used to point the reader of a remark to the appropriate point in a text.

The final use of comments is to mark changes made for layout reasons. The following is an example of a layout comment.

```
<!--layout tux: split para -->
```

Valid Comments

Do not use `--` in comments. The use of `--` causes errors.

A. Markup Conventions

SUSE documentation uses the markup described here to create the source XML files. These files are then used to create the final document formats, such as HTML and PDF.

A.1 Use of Identifiers

1. Always use an `id` attribute in parts, chapters, appendixes, sections, figures, and examples. They are often used in other elements as well, such as tables and procedures.

2. Only use ASCII characters in IDs to avoid processing problems.

3. Insert a prefix. This prefix is determined by the element to which the id attribute belongs. See Table A.1, “Abbreviations for Different Elements in an id Attribute” (page 41).

Table A.1 *Abbreviations for Different Elements in an id Attribute*

Element	Abbreviation
appendix	app
book	book
chapter	cha
example	ex
figure	fig
glossary	glo
glossterm	gt
itemizedlist	il [a]
listitem	it
indexterm	idx [b]
orderedlist	ol[a]
part	part
procedure	pro
sect1, sect2, etc.	sec
set	set

Element	Abbreviation
step	st
table	tab
variablelist	vl
varlistentry	vle

[a] Insert an id attribute only when you also put a title element inside this list.

[b] Only use it if you create index ranges

4. Identify a chapter with short, easy-to-remember English labels, like inst_yast for a chapter or section about YaST installation.

5. Each section inherits the label of its parent plus a unique part that characterizes it. For example: sec.inst_yast might have a subsection labeled sec.inst_yast.lang. Other elements with IDs inherit the ID of the parent chapter. For example, a table in this chapter about foo could have the ID tab.inst_yast.foo.

6. Separate the prefix and the following parts with a dot (.). Do not use colons (:), semicolons (;), or other “special characters.” The underbar (_) character is allowed.

Example A.1 *Typical Examples of id Values*

```
id="sec.yast.installation"
id="cha.yast"
id="tab.yast.sources"
```

A.2 The Book Structure

SUSE follows the definition of the DocBook DTD. For more detailed descriptions about book elements, refer to the TDG sections listed in Table A.2, “Elements Used in a SUSE Book” (page 43).

Table A.2 *Elements Used in a SUSE Book*

Element	Link to TDG
`appendix`	`http://www.docbook.org/tdg/en/html/appendix.html`
`book`	`http://www.docbook.org/tdg/en/html/book.html`
`chapter`	`http://www.docbook.org/tdg/en/html/chapter.html`
`glossary`	`http://www.docbook.org/tdg/en/html/glossary.html`
`part`	`http://www.docbook.org/tdg/en/html/part.html`
`preface`	`http://www.docbook.org/tdg/en/html/preface.html`
`sect1`	`http://www.docbook.org/tdg/en/html/sect1.html`

A.3 Block Structures

Block elements include figures, tables, and sections. These elements usually create a line break before and after them.

A.3.1 Figures

`figure`
  `http://www.docbook.org/tdg/en/html/figure.html`

A figure is a formal example with a title. Figures often contain graphics or other large display elements. Frequently, they get IDs and are referenced from the text with `xref` or `link`.

`imageobject`
`http://www.docbook.org/tdg/en/html/imageobject.html`

An `imageobject` is a wrapper containing `imagedata` and its associated meta information.

`imagedata`
`http://www.docbook.org/tdg/en/html/imageobject.html`

This element points to an external entity containing graphical image data.

`mediaobject`
`http://www.docbook.org/tdg/en/html/mediaobject.html`

This element contains a set of alternative “media objects”. In DocBook V3.1, three types of external objects are defined: `videoobject`s, `audioobject`s, and `imageobject`s. Additional textual descriptions may be provided with `textobject`s.

`screenshot`
`http://www.docbook.org/tdg/en/html/screenshot.html`

A `screenshot` is a graphical environment for displaying an image of what the user might see on a computer screen. It is often used to display application screen shots, dialog boxes, and other components of a graphical user interface.

`textobject`
`http://www.docbook.org/tdg/en/html/textobject.html`

A `textobject` is a wrapper containing a textual description of a media object and its associated metadata. `textobject`s are only allowed in `mediaobject`s as a fallback option. They cannot be the primary content.

Always assign an `id` attribute to a `figure` element.

Provide an appropriate image width. Future versions may contain SVG (Scalable Vector Graphics) images.

```
<figure id="fig.foo">
  <title>A Foo Picture</title>
  <screenshot>
    <mediaobject>
      <imageobject role="latex">
       <imagedata fileref="k3b_main.eps" width="80%" format="EPS"/>
      </imageobject>
      <imageobject role="html">
       <imagedata fileref="k3b_main.png" width="75%" format="PNG"/>
      </imageobject>
    </mediaobject>
  </screenshot>
</figure>
```

A.3.2 Examples

`example`
: `http://www.docbook.org/tdg/en/html/example.html`

  `example` is a formal example with a title. Examples often contain program listings or other large, block elements. Frequently, they have IDs and are referenced from the text with `xref` or `link`.

`screen`
: `http://www.docbook.org/tdg/en/html/screen.html`

  A `screen` is a verbatim environment for displaying text that the user might see on a computer terminal. It is often used to display the results of a command.

Always insert an `id` attribute inside an `example` element.

Be careful with the `screen` element: Always begin the code after the start tag and omit whitespaces before the end tag. Do not indent the code. Always start in the first row.

```
<example id="ex.foo">
  <title>A clever presentation of <command>foo</command></title>
  <screen>#!/bin/bash
echo "Hello foo"</screen>
</example>
```

If you need line numbering, insert the attribute `linenumbering` with the value `numbered`. Do not use hardcoded line numbers in the `screen` environment.

```
<example id="ex.foo">
  <title>A clever presentation of <command>foo</command></title>
  <screen linenumbering="numbered">#!/bin/bash
echo "Hello foo"</screen>
</example>
```

A.3.3 Callouts

`callout`
: `http://www.docbook.org/tdg/en/html/callout.html`

  A `callout` is a visual device for associating annotations with an image, program listing, or similar figure. Each location is identified with a mark and the annotation is identified with the same mark. This is somewhat analogous to footnotes in text.

`calloutlist`
: `http://www.docbook.org/tdg/en/html/calloutlist.html`

  A `calloutlist` is a list of annotations or descriptions. Each `callout` points to the area on a `screen` that it augments.

`co`
: `http://www.docbook.org/tdg/en/html/co.html`

  A `co` identifies (by its location) a point of reference for a callout.

Callouts do not stand alone. They are only useful in combination with `screen`. At the moment, `screenco` and `arearef` are not supported by `xsltproc`.

```
<screen>
color white/blue black/light-gray   <co id="co.col"/>
default 0   <co id="co.def"/>
timeout 8   <co id="co.time"/>
gfxmenu (hd0,0)/boot/message

title SUSE LINUX 9.1.42
    kernel (hd0,0)/boot/vmlinuz root=/dev/hda1 vga=0x31a selinux=0
    splash=silent resume=/dev/hda2 desktop elevator=as  showopts
    initrd (hd0,0)/boot/initrd

title Diskette
```

```
    root (fd0)
    chainloader +1</screen>

     <calloutlist>
      <callout arearefs="co.col">
       <para>Here, define the colors used by the
 <application>GRUB</application> menu.
       </para>
      </callout>
      <callout arearefs="co.def">
       <para>Define the default option.
       </para>
      </callout>
      <callout arearefs="co.time">
       <para>
 Select a timeout after which booting will start automatically.
       </para>
      </callout>
     </calloutlist>
```

A.3.4 Procedures

`procedure`

`http://www.docbook.org/tdg/en/html/procedure.html`

A `procedure` encapsulates a task composed of `step`s (and possibly `substep`s). Procedures are performed sequentially.

`step` , `substep`

`http://www.docbook.org/tdg/en/html/step.html`

A step identifies a unit of action in a `procedure`. If careful internal procedures are required for some steps, you can embed `substep`s in a `step`. Embedded `substep`s contain `step`s, so that `substep`s can be nested to any depth.

```
<procedure>
  <title>A Good Procedure</title>
  <step>
    <para>Get up from your bed</para>
  </step>
  <step>
    <para>Walk to the kitchen</para>
  </step>
  <step>
    <para>Make coffee or tea</para>
  </step>
</procedure>
```

Procedure A.1 *A Good Procedure*

1. Get up from your bed.
2. Walk to the kitchen.
3. Make coffee or tea.

A.4 Inline Structures

Inline elements include keys, menus, and commands. These elements are usually used inside text. They do not impose a line break.

A.4.1 Markup of Keys

`keycap`
  `http://www.docbook.org/tdg/en/html/keycap.html`

  The `keycap` identifies the letter or symbol printed on a physical key on a computer keyboard.

`keycombo`
  `http://www.docbook.org/tdg/en/html/keycombo.html`

  For actions that require multiple keystrokes, mouse actions, or other physical input selections, the `keycombo` element provides a wrapper for the entire set of events.

Mark up keys with the element `keycap`:

```
<para>
  Press <keycap>F2</keycap> to ... Use the <keycap
  function="control"/> key.
</para>
```

If you mark up general keys, like control, backspace, or enter, use the `function` attribute. Possible attribute values for `function` are `alt`, `backspace`, `command`, `control`, `delete`, `down`, `end`, `enter`, `escape`, `home`, `insert`, `left`, `meta`, `option`, `other`, `pagedown`, `pageup`, `right`, `shift`, `space`, `tab`, and `up`.

If you need key combinations (you press two or more keys at the same time or after a short period), use `keycombo`. Use the `action` attribute to distinguish between different actions:

`press` (default action)
: If you omit the `action` attribute, this value is used:

```
<para>
  Press <keycombo> <keycap function="alt"/>
  <keycap>F2</keycap> </keycombo> to ...
</para>
```

Press Alt + S to ...

`seq`
: Use this if you need keys pressed sequentially, for example, those in Emacs:

```
<para>
To save a file in Emacs, press <keycombo action="seq">
   <keycombo action="press">
     <keycap function="control"/>
     <keycap>X</keycap>
   </keycombo>
   <keycombo>
     <keycap function="control"/>
     <keycap>S</keycap>
   </keycombo>
 </keycombo>.
</para>
```

To save a file in Emacs, press Ctrl + X – Ctrl + S.

`click` , `double-click` , `other` , `simul`
: These are not used in SUSE documentation.

A.4.2 GUI Menus

`guimenu`
: `http://www.docbook.org/tdg/en/html/guimenu.html`

SUSE uses this element to simplify GUI element markup. It is used for all types of menu entries and also for buttons and GUI labels.

```
<para>
     Use the <guimenu>File</guimenu> menu to reach the
     <guimenu>Print</guimenu> option.
</para>
```

Use the *File* menu to reach the *Print* option.

To mark up nested menu structures, use `menuchoice`:

```
<para>
 To exit Emacs, use <menuchoice>
   <guimenu>File</guimenu>
   <guimenu>Exit Emacs</guimenu>
  </menuchoice>.
</para>
```

To exit Emacs, use *File* > *Exit Emacs*.

A.4.3 Commands and Options

`command`

`http://www.docbook.org/tdg/en/html/command.html`

This element holds the name of an executable program or the text of a command that a user enters to execute a program.

`option`

`http://www.docbook.org/tdg/en/html/option.html`

This element identifies an argument to a command or instruction. Options may or may not be required.

Use options for a command inside `command`. Be careful about spaces:

```
<command>xmllint <option>--help</option></command>
```

```
xmllint --help
```

A.4.4 Variables

`replaceable`

`http://www.docbook.org/tdg/en/html/replaceable.html`

Content that may or must be replaced by the user

Use `replaceable` to tag contents that should be replaced by appropriate user input. For example:

Use `ls` *`my_dir`* to list the contents of the directory *`my_dir`*.

A.4.5 Directories and Filenames

`filename`

`http://www.docbook.org/tdg/en/html/filename.html`

A filename is the name of a file on a local or network disk. It may be a simple name or may include a path or other elements specific to the operating system.

Because SUSE does not distinguish between filenames and directories by adding a trailing slash to the path or with markup. `filename` is used for both directories and filenames.

```
<para>
 See file <filename>foo.xml</filename>. Look at the
 directory <filename>/etc/sysconfig</filename>.
</para>
```

See file `foo.xml`. Look at the directory `/etc/sysconfig`.

A.4.6 Referencing

`ulink`

`http://www.docbook.org/tdg/en/html/ulink.html`

The `ulink` element forms the equivalent of an HTML anchor (`<A HREF="...">`) for cross reference by a Uniform Resource Locator (URL).

`xref`

`http://www.docbook.org/tdg/en/html/xref.html`

> The xref element forms a cross-reference from the location of the xref to the element to which it points. Unlike the other cross-referencing elements, xref is empty. The processing system must generate appropriate cross-reference text for the reader.

References to man pages and info pages should generally be made in the format “the man page of command” or “the info page of command”. In a situation where the category of the page is needed, add “(man 9 command)” to the end.

See the man page of xmllint (man 1 xmllint).

Use ulink to reference to an URL:

```
<para> See <ulink url="http://www.suse.de"/>. See <ulink
     url="http://www.novell.de">Novell Deutschland</ulink>.
```

See http://www.suse.de. See Novell Deutschland [http://www.novell.de].

If you need references to another chapter, section, etc., in the same documentation set, use xref:

```
<para> A link to this section: <xref
     linkend="sec.markup.inline.referencing"/> </para>
```

A link to this section: Section A.4.6, “Referencing” (page 51)

If you need a different text for referencing purposes, but you cannot modify the title, use the xreflabel attribute:

```
<sect1 id="sec.foo" xreflabel="Hello foo">
  <title>Foo</title>
  <para>...</para>

</sect1>
  ...
<para>See <xref linkend="chap.foo"/>...</para>
```

In the above example, the <xref/> is replaced by the text Hello foo.

A.4.7 Emphasis

`emphasis`

> `http://www.docbook.org/tdg/en/html/emphasis.html`
>
> This element provides a method for indicating that certain text should be stressed in some way. However, placing emphasis with careful wording is preferred to using markup.

Use bold text with care.

```
<para>
 This is <emphasis>strong</emphasis> word.
 This is a  <emphasis role="bold">very strong</emphasis> word.
</para>
```

This is a *strong* word.
This is a **very strong** word.

A.4.8 Comments and Remarks

`remark`

> `http://www.docbook.org/tdg/en/html/remark.html`
>
> The `remark` element is designed to hold remarks, such as editorial comments, that are useful while the document is in the draft stage, but are not intended for final publication.

`<!--An XML comment-->`

> XML comments are used to hold remarks. They are invisible in your publication.

Use `remark`s for comments that you can switch on or off in HTML or PDF. Place a `remark` before or after the `para` to comment. Otherwise you would insert unwanted line breaks.

```
<remark>A small remark</remark>
```

Use XML comments if you give translators or other editors some hints that should only be visible in the code, but not in the final publication—even for a draft state.

```
<para> <!-- Is this true? Please check! --> If you need X,
     you must install Y before you can use it. </para>
```

If you need X, you must install Y before you can use it.

Be short and clear with your comments.

A.4.9 Literals

`literal`

   `http://www.docbook.org/tdg/en/html/literal.html`

   `literal` is specific data taken literally from a computer system. It is similar in some ways to `userinput` and `computeroutput`, but is somewhat more general a classification.

```
<para>
  If you need a comment, insert <literal>#</literal> characters.
<para>
```

B. GNU Free Documentation License

Version 1.2, November 2002

B.1 PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

B.2 APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To

"Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

B.3 VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

B.4 COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

B.5 MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

B.6 COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

B.7 COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

B.8 AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

B.9 TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

B.10 TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

B.11 FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

B.12 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.
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, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.