



SUSE Documentation Style Guide

SUSE Documentation Style Guide

This guide provides answers to writing, style and layout questions commonly arising when editing SUSE documentation. The GeekoDoc and DocBook markup reference in this guide will help you choose the right XML element for your purpose. Following this guide will make your documentation more understandable and easier to translate.

Publication Date: 12/12/2024

<https://documentation.suse.com> 

Copyright © 2007– 2024 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

For SUSE trademarks, see <https://www.suse.com/company/legal/> . All other third-party trademarks are the property of their respective owners. Trademark symbols (®, ™ etc.) denote trademarks of SUSE and its affiliates. Asterisks (*) denote third-party trademarks.

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

Contents

1	Writing technical documentation	1
2	Documentation content	2
3	Names of example items	3
3.1	Domains	3
3.2	Host names	3
3.3	IPv4 addresses	3
3.4	IPv6 addresses	3
3.5	Users	4
4	Outline of a manual	5
4.1	Books	5
4.2	Articles	8
5	Writing for the Web	9
5.1	Topical structure	9
5.2	Writing SEO-friendly content	9
5.3	Writing for a global audience	11
6	Language	13
6.1	Abbreviations	13
	Acronyms	13 • Latin abbreviations 14 • Units of measurement 14
6.2	Biases and inclusiveness	14
6.3	Capitalization of headings and titles	14
	Most titles: sentence-style capitalization	14 • Document titles: title-style capitalization 15

6.4	Commas	15
6.5	Contractions	15
6.6	Dashes	15
6.7	End of sentence punctuation	16
6.8	File and directory names	16
6.9	Headings	17
6.10	Hyphens	17
6.11	Lists	17
6.12	Numbers and measurements	18
6.13	Possessives	18
6.14	Prefixes	18
6.15	Quotations	18
6.16	Semicolons	19
6.17	Sentence structure	19
6.18	Slashes	19
6.19	Tense	20
6.20	Tone and voice	20
6.21	Trademarks	20
6.22	User interface items	21
7	Structure and markup	22
7.1	Admonitory and advisory paragraphs	22
7.2	Application names	24
7.3	Callouts	24

7.4	Command-line input and command-line output	26
	Commands	27 • File names 27 • Literals 28 • Placeholders 28 • Prompts 28 • Screens 29 • Variable names 30
7.5	Cross-references	30
7.6	Emphasis	31
7.7	Examples	32
7.8	External links	33
7.9	External links to SUSE documentation	34
7.10	Figures	36
	Graphics	37 • Screenshots 37
7.11	Glossaries	39
7.12	Identifiers	40
7.13	Lists	43
	Itemized lists	44 • Ordered lists 45 • Variable lists 46
7.14	Keys and key combinations	47
7.15	Outline levels and sectioning	49
7.16	Procedures	49
7.17	Products	51
7.18	Profiling	51
7.19	Questions and answers	52
7.20	References to other external resources	54
7.21	Tables	55
7.22	User interface items	57

8	DocBook tags	59
8.1	Using DocBook tags	59
	Meta elements	59 • Structure elements 61 • Block elements 63 • Inline elements 65
8.2	Using ITS tags	68
9	Managing documents	69
9.1	Remarks	69
9.2	XML comments	69
9.3	Entities	70
	Common types of entities	70 • Using entity files 71
9.4	XInclude elements	73
10	Formatting XML	74
11	Working with AsciiDoc	77
12	Working with Smart Docs	78
12.1	Using ITS tags with assemblies	78
12.2	Revision history	83
A	Terminology and general vocabulary	85
A.1	Terminology	85
A.2	General vocabulary	114
B	Contributors	118
C	GNU Free Documentation License	119

1 Writing technical documentation

Technical writing has certain characteristics that make it different from other types of writing. Its objective is to provide readers with complex information and comprehensive answers they are searching for. The content should be well-structured, clear and concise. Effective technical documentation is straightforward, detailed and focused on problem-solving, and there is a specific workflow for its creation:

Defining the target audience

Adjust tone, style and technicality of the text based on the intended audience. Keep in mind that not all facts that seem obvious to you will be obvious to your readers.

Researching a topic

Start with research on the information that is relevant to the target audience. Receive essential input from issue tracking systems like GitHub, and project management tools.

Writing about a topic

Start writing at an early stage, even if you have not finished your research yet. Prepare a draft document and discuss it with subject-matter experts.

Getting reviews

In a first review of your text, identify and fix the most obvious issues like typos, unfinished sentences, etc. After self-review, ask for a technical review by dedicated specialists. A technical review uncovers technical or factual errors like missing or misspelled package names, wrong commands or forgotten options.

Request a peer review which can improve your text and detect any structural problems or logical traps. You can then do a spell check, link check and style check with the DAPS tool. Finally, ask for a linguistic review that tackles language issues, typos, inconsistencies, and style guide compliance.

For more information on how to produce meaningful content that will rank high on the Web, see [Chapter 5, Writing for the Web](#) and [Section 5.2, "Writing SEO-friendly content"](#).

2 Documentation content

When selecting what to document, keep to the following guidelines:

- Do not promise future features. Only document features that exist already or that will be finished before the document is first published.
- In some cases, it is appropriate to warn of scheduled future changes, such as feature removals.
- Documentation concerning unsupported features should be an exception. When documenting an unsupported feature (usually a *technology preview*), explain the support status before going into detail about the feature.

3 Names of example items

This section summarizes conventions for creating generic names for objects in documentation. Most of the following names are covered through entities. To check their spelling, refer to the Doc Kit repository at <https://github.com/openSUSE/doc-kit/tree/main/entities>. See also *Section 9.3, “Entities”*.

3.1 Domains

Use <http://www.example.com> and <http://www.example.org> as example domains. Both domains were registered for use in documentation.

3.2 Host names

Use objects of the solar system: For the most important system, use sun. For other systems, use the names of planets such as earth or mars.

3.3 IPv4 addresses

Use addresses from the class C subnet 192.168.255.0 for examples. That is, replace the final 0 with any integer between 0 and 255. To create examples using a larger setup, use addresses from the private network ranges. For more information, see http://en.wikipedia.org/wiki/Private_network.

3.4 IPv6 addresses

Use addresses from the subnet 2001:0db8::/32 for examples. That is, after the 2001:0db8: prefix, add six four-digit numbers, each separated by a colon on both sides. Each of the hexadecimal digits may have a value between 0 and f. A valid example URL is 2001:0db8:0123:4567:89ab:cdef:1234:5678. For more information, see http://en.wikipedia.org/wiki/IPv6_subnetting_reference.

3.5 Users

For example users, use free-software mascots, such as Tux (Linux Kernel), Wilber (The GIMP), Geeko (SUSE), Foxkeh (Firefox), Konqi (KDE), or Duke (Java).

4 Outline of a manual

Maintain a consistent structure within your documents. The structure can vary between different books, articles or projects, but the most common parts of the document structure are documented here.

4.1 Books

For books, use a document structure that includes the following elements, in that order:

1. a preface
2. chapters, split into sections
3. *(optional)* a glossary
4. *(optional)* appendixes

For books with many chapters, create parts at the outline level above chapters.

Title page and imprint

Both title page and imprint are created automatically, but depend on information being present in the book.

- **Title.** Work with the product manager to define the book title. The book title should not contain the product name and version.
- **Product name and product version.** Work with the product manager to find the correct product name and version number. Mark this information up with `<product-name/>` and `<productnumber/>`, respectively.
- **Documentation version or revision information.** Optionally, use the `<releaseinfo/>` element to mark up version or revision numbers of the documentation itself.
- **Copyright notice.** Use the standard copyright notice reproduced below. Change the starting year of the copyright protection to the current year.

EXAMPLE 4.1: STANDARD COPYRIGHT NOTICE

```
<legalnotice>
```

```

<para>
  Copyright &copy; [starting year]&ndash;<?dbtimestamp format="Y"?>
  SUSE LLC and contributors. All rights reserved.
</para>
<para>
  Permission is granted to copy, distribute and/or modify this document
  under the terms of the GNU Free Documentation License, Version 1.2 or
  (at your option) version 1.3; with the Invariant Section being this
  copyright notice and license. A copy of the license version 1.2 is
  included in the section entitled <quote>GNU Free Documentation
  License</quote>.
</para>
<para>
  For &suse; trademarks, see
  <link xlink:href="http://www.suse.com/company/legal/">.
  All other third-party trademarks are the property of their respective
  owners. Trademark symbols (&reg;, &trade; etc.) denote trademarks
  of &suse; and its affiliates. Asterisks (*) denote third-party
  trademarks.
</para>
<para>
  All information found in this book has been compiled with utmost
  attention to detail. However, this does not guarantee complete
  accuracy. Neither SUSE LLC, its affiliates, the authors nor the
  translators shall be held liable for possible errors or the
  consequences thereof.
</para>
</legalnotice>

```

Abstract

Use an abstract to summarize the information provided in a book, article, chapter, or set in 70–150 characters. Do not use lists, images or examples in an abstract.

EXAMPLE 4.2: AN ABSTRACT

Perform an upgrade of a SUSE Linux Enterprise system to a new major version in environments which do not allow for standard booting of the installation.

Table of contents

The table of contents is generated automatically.

Preface

Include a brief overview of the content of a manual, related manuals and typographical conventions. The preface can also contain information about its target audience.

Parts

If you are writing a book with many chapters, create parts at the outline level above chapters. Parts should contain at least three chapters. Keep part titles clear and concise. Often a single noun is enough. Typical part titles are *Installation* or *Network*.

Chapters

Chapter titles should not contain product version numbers which affect the output of data analytics. Chapters typically consist of the following elements (appendixes should be regarded an exception):

- **Abstract.** In an abstract, summarize the topic instead of summarizing the outline. See also [Abstract](#).
- **Introduction.** Provide introductory information directly after the abstract. Do not place it in a separate section.
- **Sections.** Structure the detailed information, so readers can skim the text. Create sections for every major task, such as installing, configuring, monitoring, and administering. If helpful, split sections into subsections. Avoid nesting deeper than three levels of sections.
Start sections with an introductory paragraph outlining the focus of the section. If the section describes a sequential task, add a procedure description, as discussed in [Section 7.16, "Procedures"](#). Steps of a procedure can contain a cross-reference to subsections where topical background is provided and an action is explained in detail. See also [Section 7.5, "Cross-references"](#).
- **Troubleshooting.** In this section, collect common mistakes and problems. The section should always be named *Troubleshooting*. Use the DocBook element `<qandaset/>` (a Question and Answer section) to mark up *Troubleshooting* problems. In case you want to describe solutions to more than ten problems, add topical subsections (`<qandadiv/>` elements) below the *Troubleshooting* section.
- **More information.** In a section named *More information*, collect Web links to all sources of information that might prove helpful in a given context. Follow the general referencing guidelines in [Section 7.5, "Cross-references"](#) when creating such sections.

Glossary

The optional glossary contains important terms in alphabetical order and provides a brief explanation for each. For more information on creating lists of terms, see [Section 7.11, "Glossaries"](#).

Contributors

Writing documentation is a collaborative effort. To give credit to all contributors, add an appendix to the guides which points to the *Contributors* page for the respective GitHub repository. For an example, see [Appendix B, Contributors](#).

For articles and small documents (such as SUSE Best Practices) whose content is created and maintained by five or fewer contributors, all of whom are from outside the documentation team, credit the contributors on the title page.

The above are a minimum. In addition, make sure that the contributors appendix is compliant with the document license.

4.2 Articles

For articles, use a document structure that includes the following elements, in that order:

1. introduction
2. sections, split into subsections
3. *(optional)* a glossary
4. *(optional)* appendixes

5 Writing for the Web

Create engaging and informative content that helps your audience and is optimized for the Web.

5.1 Topical structure

The most important thing to do with your Web copy is to help users get answers to their questions as soon as possible. To achieve this, we recommend the modular approach of topic-based authoring where documentation is created and maintained in chunks, subsequently referred to as topics.

Topics have the sole purpose of supporting users in their tasks. Each topic focuses on one specific subject and has one distinct purpose. We write topics in a way that they can stand alone as well as in context with other topics. They should also be reusable in different contexts.

Recommendations:

- Put your most important information first. Users typically decide if they are going to stay on your page within 3–5 seconds. Make sure your copy helps users understand the big picture right away.
- Write for scanners. Help readers find the answer to their question. Create headlines that are clear and to the point. Break long headlines into a heading and sub-head. Ask yourself: Is it easy to see the benefit of the page with a quick glance?
- Organize content logically. Layer and break up your content into sections to help users find answers at a glance.

5.2 Writing SEO-friendly content

SEO, or Search Engine Optimization, is the strategy to attract organic (unpaid) traffic that originates from online search and refers to visitors landing on the Web site. When it comes to SEO, content is key. Here are some insights to help you create and design the content that will rank high in search engines.

Brainstorm and research keywords

To learn what type of content search engines deem best for a specific keyword, search the phrases you want to rank for. Bear in mind the approved SUSE documentation terminology.

At the same time, do not stuff your content with keywords. The keyword ratio for an article should be 2–4%, that is, 8–10 keywords per 500 words.

Structure your content in a way that is easy for users to scan

Structuring your headings in a hierarchy can make a larger content piece easily scannable while helping search engines understand the context of your content. For example:

```
<H1> Dealing with Boot and Installation Problems
<H2> Problems Booting
  <H3> Solution 1
  <H3> Solution 2
<H2> Problems Installing
  <H3> Solution 1
  <H3> Solution 2
```

Create concise and meaningful title tags and meta descriptions

Search engines pull the title tag, or meta title, from H1, and the meta description from the abstract. The number of characters is limited, and the recommended length for meta titles is 55–65 characters. However, they must not be shorter than 29 characters. Be sure to optimize your headings wherever appropriate by integrating keywords, such as product names, into them. The optimal length for meta descriptions is 145–155 characters.

Product names are known for their extra length, so effective page titles and meta descriptions may require the use of abbreviations. From the search engine perspective, it can be helpful to mention the abbreviation within the content by adding the abbreviation in parenthesis after the first use of the term.

Link from (and to) your content

Links are critical to establishing the authority and relevance of your content. The two most important types of links are:

1. Internal links lead to and from other pages within the same domain and help establish the relationship between two pieces of content. Internal linking helps search engines discover new pages on the Web site and index them.
2. Inbound links lead to your content piece from a different domain (for example, SAP.com to SUSE.com).

External links to high-quality, creditable Web sites help increase the validity of your own Web site. The better the links, the higher the page ranks in search results.

Optimize images by adding alt text

Be sure to add alternative text whenever you use images. This practice allows search engines to understand the context of your images and index them more accurately. It also improves the accessibility of the Web site.

5.3 Writing for a global audience

Remember that every document is a potential candidate for localization (translation). Make sure the document's original English content is correct and clear. Simplicity, clarity and direct prose are essential.

Recommendations:

- Keep sentences short. Shorter sentences help translators and target audience to better understand the content.
- Be consistent. Stick to the terminology and use the same sentence structure for similar content. Use the same sentences for repetitive texts. This helps to improve the translation memory leverage.
- Use proofreading and review options. Have your content reviewed to detect misunderstandings in advance.
- Keep it clear. Make clear statements and avoid “should,” “could” and similar unprecise words.
- Mark non-translatable text. Use tags as defined in [Chapter 8, DocBook tags](#) or use the ITS tag feature to mark all content that should not be translated (for more information, see [Section 8.2, “Using ITS tags”](#)). Use entities for product names, example names, etc. Always mark code as such so that it is not translated.
- Write for the world (if possible). Do not use country-specific words and examples. Use common international examples instead.
- Use only as many graphics as needed. As each graphic or screenshot needs to be localized as well, keep it to the minimum.
- Do not break sentences. Do not use hard breaks within a sentence.
- Do not break your sentence with lists. For example, do not structure the phrase like this:

You can use the following commands:

```
-a  
-z  
-b  
to start the system update.
```

Keep the sentence together instead:

```
To start the system update, you can use the following commands:  
-a  
-z  
-b
```

6 Language

We write documentation in American English. Where spelling differences exist between American and British English, use the American English variant. For verbs ending in either *-ise* or *-ize* (like *localise/localize*), use the *-ize* variant.

When in doubt about the spelling or usage of a word, first see [Appendix A, Terminology and general vocabulary](#). If the usage of a word is not regulated there, use the preferred spelling from <https://www.merriam-webster.com/> [↗](#) (<https://www.m-w.com/> [↗](#) for short).

The correct spelling of SUSE product names is listed in the terminology table ([Appendix A, Terminology and general vocabulary](#)) and in the entities file of the Doc Kit repository at <https://github.com/openSUSE/doc-kit/blob/main/entities/generic-entities.ent> [↗](#). If a product name is not listed in either spot, refer to the official SUSE Products (<https://www.suse.com/products/>) [↗](#) page and the Marketing department. Make sure to not use articles in front of product names.

When in doubt about a style rule, see *The Chicago Manual of Style*, 15th Edition.

6.1 Abbreviations

Avoid using abbreviations, especially unusual ones. Avoid creating plurals of abbreviations, unless the abbreviation is an acronym or initialism.

6.1.1 Acronyms

Introduce acronyms by providing the expansion in parentheses after the acronym. Sometimes chapters and parts are used across multiple documents. Therefore, provide the expansion of an acronym at least once per chapter.

However, do not use headlines to introduce an acronym. Headlines or captions must not contain both an acronym and its expansion. If a term is commonly written as an acronym, use the acronym in the title. When mentioning the term for the first time in the following text, use its expanded form. All following occurrences of the term in this chapter should then use the acronym.

Create plural forms of acronyms by adding a lowercase “s”. For example, use “CDs” and “BIOSes.” Never add an apostrophe before the “s” or “es.”

For clarity, avoid using possessive forms of acronyms. For example, do not use “XML’s specification.”

6.1.2 Latin abbreviations

Do not use Latin abbreviations. Use the full English form: for example, use “that is” instead of “i.e.”. As an exception to this rule, the abbreviation *etc.* is allowed.

6.1.3 Units of measurement

You may use abbreviations of common units of measurement. For more information about units of measurement, see [Section 6.12, “Numbers and measurements”](#).

6.2 Biases and inclusiveness

Do not artificially limit your audience by excluding or offending members of it.

Avoid indicating gender in your documentation. If possible, use plural to allow use of “they” as the pronoun. Otherwise, use “he or she.”

SUSE supports the Inclusive Naming Initiative which aims to help avoid harmful language. When making language choices for documentation, check the initiative's [Evaluation Framework \(https://inclusivenaming.org/language/evaluation-framework/\)](https://inclusivenaming.org/language/evaluation-framework/) and its “Word lists.”

The SUSE official terminology database, [TermWeb \(https://suse.termweb.eu/\)](https://suse.termweb.eu/), also contains inclusive naming recommendations.

For more information about avoiding gender bias, see *The Chicago Manual of Style*, 5.43. For information about names of example items, see [Chapter 3, Names of example items](#).

6.3 Capitalization of headings and titles

6.3.1 Most titles: sentence-style capitalization

Sentence-style capitalization is the most common capitalization used in SUSE documentation. When using sentence-style capitalization, only proper nouns and the first letter of the first word of a phrase are capitalized. Apply sentence-style capitalization to all running text and all types of headings and titles that are part of the document content. An example for sentence-style capitalization is “Ceph core components.”

6.3.2 Document titles: title-style capitalization

For document titles, such as book, article, and set titles, use title-style capitalization. This capitalization style is explained in *The Chicago Manual of Style*, 8.167. A simplified version of these rules is below:

1. Capitalize the first and the last word.
2. Write articles in lowercase. Articles are: *the*, *a*, and *an*.
3. Write prepositions in lowercase unless they are used with a verb (“Logging In”) or in a noun (“The On Button”). Prepositions are, for example: *up*, *in*, *of*, *through*, and *between*.
4. Write certain conjunctions in lowercase: *and*, *but*, *for*, *nor*, and *or*.
5. Write *as* and *to* in lowercase.
6. Capitalize everything that is not mentioned above.

Examples for title-style capitalization are “Deployment Guide” (book title) or “Kernel Module Packages for SUSE-Based Distributions” (article title).

6.4 Commas

Use commas to separate elements in a series of three or more elements, but do not put a comma before the conjunction in most simple series. For example, “Find basic information about how to register your system, modules and extensions.” Use commas around phrases like *for example* and *that is*. Introductory phrases at the beginning of a sentence are normally followed by a comma. For example, “Before using YaST Online Update, configure a network connection.”

6.5 Contractions

Do not use contractions (such as “don’t”), unless you are purposefully writing a casual document.

6.6 Dashes

Use en dashes (–) between numbers in a range in tables and figures.

For punctuation, use em dashes (—). Do not surround em dashes with spaces. Use em dashes sparingly.

6.7 End of sentence punctuation

End sentences in a period. Avoid using exclamation marks. Restrict question marks to question and answer sections.

6.8 File and directory names

Under Linux, objects like directories, printers, or flash drives are all considered files. Therefore, the naming and markup conventions are the same for “drives” (for example, hard disks, CD-ROM drives), directories, or files.

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

- In general, use forward slashes (/) to separate nested directory or file names. If you are describing actions performed on Windows* systems and within a Windows-native file system, use backward slashes (\) instead.
- In general, when giving absolute paths, always start with a leading slash to indicate the root of the file system. If you are describing actions performed on Windows systems and within a Windows-native file system, do not add a leading slash to absolute paths.
- When referencing a directory name, add a trailing slash. This helps distinguish between directory names (for example, `/etc/YaST2/`) and file names (for example, `/etc/YaST2/control.xml`). For less experienced Linux users, it might be helpful to specify in the running text if it is a file, device, or directory. For example: “In the `/etc/hosts/` directory, do the following.”

Most Linux file systems are case-sensitive. Use capitals exactly as they appear in the file system. For more information about markup aspects, see [Section 7.20, “References to other external resources”](#) and [Section 7.4.2, “File names”](#).

When it is necessary to refer to file extensions, such as in compound words like “PDF file,” always capitalize the extension.

6.9 Headings

When writing a descriptive section, use a noun-based heading title, for example, “Concepts of Software.” When writing a task-orientated section, use a verb in gerund, for example, “Installing Software.”

Keep headings short and simple. Do not use both an acronym and the expanded form in a heading. Make sure that headlines in a chapter follow the same pattern.

For advice on how to nest sections, refer to [Section 7.15, “Outline levels and sectioning”](#).

6.10 Hyphens

Generally, hyphens are used as joiners for two or more words that form a single concept and function together as a compound modifier before the noun. If the noun comes first, the hyphen is not added. For example, “the list in the upper-left corner” but “place the list in the corner in the upper left.”

There are technical guidelines to help you choose whether to use or not to use a hyphen.

Add the hyphen when:

- The last letter of the prefix and the first letter of the word are the same (“shell-like”). However, double-e combinations usually do not get a hyphen: “preempted,” “reelected.”
- The words begin with the prefixes *self-*, *ex-* (that is, “former”), and *all-*: “self-assigned,” “ex-service,” “all-data.”

Do not use the hyphen when:

- The prefix and the following word start with a consonant (“subpackage”).
- The two-word phrase includes the adverb *very* and all adverbs ending in *-ly*: “a very good time,” “an easily remembered rule.”

Many combinations that are hyphenated before a noun are not hyphenated when they occur after a noun. For example: “This is the up-to-date version” and “The calendar is up to date.”

6.11 Lists

For information about creating lists, see [Section 7.13, “Lists”](#).

6.12 Numbers and measurements

Write the integers zero through nine as words. Use numerals for all other numbers.

When the unit of a measurement is abbreviated, always use numerals for the number. In measurements, add a non-breaking space (` `) between the numeral and its corresponding unit abbreviation. Use the % sign when paired with a number, with no space.

For more information, see *The Chicago Manual of Style* 9.6 and 9.16.

6.13 Possessives

Do not use possessives of acronyms and trademarked terms. Avoid possessives of inanimate objects.

6.14 Prefixes

Add a hyphen after the prefix to prefixed words only if you foresee misunderstandings. For example, there is a difference in meaning between “recreate” and “re-create.”

For more information about using hyphens, see [Section 6.10, “Hyphens”](#).

6.15 Quotations

Use quotations to quote from sources, such as books. In all other cases, do not use quotation marks:

- For computer input, computer output and user interface elements, use different markup. See also [Section 7.4, “Command-line input and command-line output”](#), [Section 7.22, “User interface items”](#), and [Section 8.1, “Using DocBook tags”](#).
- Use `<emphasis>` to call attention to new words or phrases, for example, “using so-called *target units*,” to use words in a non-standard way, for example, “packages can get in an *orphaned* state,” and to refer to a word or term itself, for example, “The word *processor* came into use around 1910.”
- Do not use quotation marks to indicate irony. Avoid irony in technical writing. See also [Section 6.20, “Tone and voice”](#).

To create quotations, use the `<quote/>` element. This element is easier to localize than hard-coded quotation marks. During processing, localized quotation marks are added automatically. Avoid using single quotation marks.

The period and the comma always go within the quotation marks, as illustrated in [Example 6.1, “Quote”](#). The dash, the semicolon, the colon, the question mark and the exclamation mark go within the quotation marks when they apply to the quoted matter only. They go outside when they apply to the whole sentence.

EXAMPLE 6.1: QUOTE

| “Suds may froth,” the sign reads.

6.16 Semicolons

Avoid using semicolons to join sentences. You may use semicolons in place of commas in very complicated series.

6.17 Sentence structure

Form clear and direct sentences with 25 words or less. Avoid complicated clauses. Make sure that the relationship between subject, verb, and object is clear. Avoid joining sentences with semicolons. Avoid ending sentences with prepositions.

Avoid using parentheses. Where they are necessary, move them to the end of the sentence. Never nest parentheses.

Always let the reader know the objective of an action before describing the action itself. As an example, write: “To save the settings, click *OK*.”

6.18 Slashes

Do not use slashes except when they are part of a standard technical term, such as *TCP/IP* or *client/server*. Do not add spaces on either side of a forward slash.

6.19 Tense

Use the simple present tense. Apply the simple present tense even to sentences with “if” or “when” clauses and to prerequisites of an action. For example, “If this happens, go there.” or “Glibc is installed.”

6.20 Tone and voice

Maintain a professional tone. Do not use contractions, except in casual documents. Do not use humor. Be honest and avoid absolutes and exaggerations, but focus on positive aspects.

Use the second person (“you”) to refer to the reader. Normally, the reader is the user or administrator who performs the actions described. For example, “To install all officially released patches that apply to your system, run **zypper patch**.” Do not overuse “you” and “your.” It is often implied who you are addressing in the instructions. For example, instead of “Install *package* on your system,” just say “Install *package* on the system.”

Where possible, use active voice. If there is no emphasis on the object of the verb or if the performer of the action is unknown, use passive voice. “A Samba server must be configured in the network” is an example of the proper use of passive voice. The emphasis is on the server, not on the person configuring it.

When giving a recommendation, start with “We recommend.” Do not use passive phrasings like “It is recommended.”

To refer to other parts of the document, start with “For more information (about), see.”

6.21 Trademarks


Most products referenced in the documentation are trademarked. Follow these rules when dealing with these terms:

- Never use trademarks in headings.
- Only use the ®, ™ or SM marks for SUSE products.
- Use an * (asterisk) for all service marks or trademarks of third-party companies. This acknowledges the service mark or trademark of the other company. It also protects SUSE if the protection of the brand changes in any way.

For more information about markup aspects, see [Section 7.17, “Products”](#).

6.22 User interface items

When referring to labels of user interface items, do not include ending punctuation such as `...` or `:`. Whenever possible, refer to user interface items without identifying them as any special type of element. For example, use “click *OK*” rather than “click the *OK* button.” However, complex dialogs may require more specific wording.

When referring to UI labels, capitalize them exactly as in the UI itself. Software created at SUSE (such as YaST or Uyuni) should use sentence-style capitalization. If it does not, you can make aware the developers of that software. For more information about sentence-style capitalization, see [Section 6.3.1, “Most titles: sentence-style capitalization”](#) and the [SUSE Product Brand guide \(https://productbrand.suse.com/writing/conventions-and-rules\)](https://productbrand.suse.com/writing/conventions-and-rules) .

For more information about markup for UI labels, see [Section 7.22, “User interface items”](#).

7 Structure and markup

This chapter contains instructions on using the correct DocBook markup to create consistent and legible documents, and structuring the content in the way that it effectively helps readers find answers to their queries.

In the Doc Kit repository, you can see examples of how [books](https://github.com/openSUSE/doc-kit/blob/main/docbook5-book/book-example_en.pdf/) and [articles](https://github.com/openSUSE/doc-kit/blob/main/smart-doc/article-example_en.pdf/) are rendered in our documentation.

SUSE uses the GeekoDoc Relax NG schema which is compatible with DocBook 5.2. For more information about the XML elements described here, see the *DocBook 5.2: The Definitive Guide* sections listed in [Table 7.1, “Important elements”](#).

TABLE 7.1: IMPORTANT ELEMENTS

Element	Web link
<u><appendix/></u>	https://tdg.docbook.org/tdg/5.2/appendix.html
<u><book/></u>	https://tdg.docbook.org/tdg/5.2/book.html
<u><chapter/></u>	https://tdg.docbook.org/tdg/5.2/chapter.html
<u><glossary/></u>	https://tdg.docbook.org/tdg/5.2/glossary.html
<u><part/></u>	https://tdg.docbook.org/tdg/5.2/part.html
<u><preface/></u>	https://tdg.docbook.org/tdg/5.2/preface.html
<u><sect1/></u>	https://tdg.docbook.org/tdg/5.2/sect1.html

7.1 Admonitory and advisory paragraphs

To make readers aware of potential problems and recent changes, or to give them tips, use an admonition element. Avoid using more than one admonition per page of PDF output.

- `<warning/>`. Use these elements to warn of security issues, potential loss of data, damage to hardware, or physical hazards. Warnings must always precede the action to which they apply.
- `<important/>`. Use these elements to give vital information.
- `<tip/>`. Use these elements to introduce guidelines or give tips.
- `<note/>`. Use these elements to highlight software version differences.

Follow these rules when writing admonitions:

- Add a `<title/>` to admonitions. In the title, state the subject of the admonition and, in the case of a `<warning/>`, also the source of danger.
- `<warning/>` or `<important/>`: In the first paragraph, clearly state possible consequences of ignoring the danger.
- `<warning/>` or `<important/>`: In the second paragraph, explain how to avoid the danger. If there are multiple ways to avoid a danger, use an unordered list. If fewer than five consecutive steps must be taken to avoid a danger, use an ordered list. If more than five consecutive steps need to be taken, use a cross-reference to another part of the documentation.

EXAMPLE 7.1: AN EXAMPLE OF A WARNING (SOURCE)

```
<warning>
<title>Do not interrupt creation of file systems</title>
<para>
  Creating a file system can take multiple hours.
  Interrupting this process will result in a corrupt file system and an
  unusable installation.
</para>
<para>
  Always wait until formatting has finished.
</para>
</warning>
```



Warning: Do not interrupt creation of file systems

Creating a file system can take multiple hours. Interrupting this process will result in a corrupt file system and an unusable installation.

Always wait until formatting has finished.

7.2 Application names

When referring to an application, add a `<phrase role="productname"/>` element around it:

```
<phrase role="productname">LibreOffice</phrase> is an office suite.
```

This will not result in a visual change but disables hyphenation in browsers. This markup side-steps hyphenation issues of CamelCased names.

7.3 Callouts

Callouts allow annotating examples, such as configuration files or commands with many options. Add `<co/>` elements directly after the part of a screen that you want to annotate. Do not try to align them above the part of a screen to annotate. Do not use more than ten callouts per example.

See also [Section 7.7, “Examples”](#).

EXAMPLE 7.2: EXAMPLE OF CALLOUTS (SOURCE)

```
<screen>color white/blue black/light-gray <co xml:id="co-color"/>
default 0 <co xml:id="co-default"/></screen>
<calloutlist>
  <callout arearefs="co-color">
    <para>
      Colors of the boot loader menu.
    </para>
  </callout>
  <callout arearefs="co-default">
    <para>
      Defines the preselected option.
    </para>
  </callout>
</calloutlist>
```

EXAMPLE 7.3: EXAMPLE OF CALLOUTS (OUTPUT)

```
color white/blue black/light-gray ❶
default 0 ❷
```

- ❶ Colors of the boot loader menu.
- ❷ Defines the preselected option.

TABLE 7.2: ELEMENTS RELATED TO `<callout/>`

Element	Web link
<code><co/></code> . Inline element to mark an area with- in a <code><screen/></code> .	https://tdg.docbook.org/tdg/5.2/co.html ↗
<code><calloutlist/></code> . Block element containing a list of descriptions for each of the marked areas.	https://tdg.docbook.org/tdg/5.2/callout- list.html ↗
<code><callout/></code> . Block element containing a de- scription of a single area marked with <code><co/ ></code> .	https://tdg.docbook.org/tdg/5.2/callout.html ↗

You can group different callouts to point to the same annotation. This helps to avoid repeating the same annotation. To do this, complete the following steps:

1. Create the `<co/>` element with the `<xml:id/>` attribute to comment on the first line.
2. On another line, use the `<xref/>` element to point to the ID from the previous step.

EXAMPLE 7.4: EXAMPLE OF A CALLOUT GROUP (SOURCE)

```
d1 = dict() <co xml:id="co-dict"/>
d2 = {} <xref linkend="co-dict"/>
l1 = list() <co xml:id="co-list"/>
l2 = [] <xref xml:id="co-list"/>
<calloutlist>
  <callout arearefs="co-dict">
    <para>A dictionary.</para>
  </callout>
  <callout arearefs="co-list">
    <para>A list.</para>
  </callout>
</calloutlist>
```

EXAMPLE 7.5: EXAMPLE OF A CALLOUT GROUP (OUTPUT)

```
d1 = dict() ❶
  d2 = {} ❶
  l1 = list() ❷
  l2 = [] ❷
```

- ❶ A dictionary.

- ② A list.

7.4 Command-line input and command-line output

When dealing with user input and system output shorter than 30 characters, format it with an inline element, such as `<command/>` or `<filename/>`. In all other cases, close the current paragraph and enclose the user input and/or system output in a `<screen/>` element. See also [Section 7.7, “Examples”](#).

When using stand-alone `<command/>` elements that are outside of a `<screen/>`, do not use `<prompt/>` elements before or within them. For more information about `<prompt/>`, see [Section 7.4.5, “Prompts”](#).

TABLE 7.3: ELEMENTS RELATED TO COMMAND-LINE INPUT AND OUTPUT

Element	Web link
<code><screen/></code> . Block element in which all characters are reproduced exactly as they are in the source of the document. See also Section 7.7, “Examples” . Can contain any of the inline elements listed in this table.	https://tdg.docbook.org/tdg/5.2/screen.html ↗
<code><command/></code> . Inline element that contains the name of an executable program or the command that a user types to execute a program. Can contain <code><replaceable/></code> elements.	https://tdg.docbook.org/tdg/5.2/command.html ↗
<code><option/></code> . Inline element that contains an argument to a command or instruction. Can contain <code><replaceable/></code> elements.	https://tdg.docbook.org/tdg/5.2/option.html ↗
<code><replaceable/></code> . Inline element that contains content that can or must be replaced by the user.	https://tdg.docbook.org/tdg/5.2/replaceable.html ↗
<code><filename/></code> . Inline element that contains the name of a directory or file. Can contain <code><replaceable/></code> elements.	https://tdg.docbook.org/tdg/5.2/filename.html ↗

Element	Web link
<code><varname/></code> . Inline element that contains the name of a variable. Can contain <code><replaceable/></code> elements.	https://tdg.docbook.org/tdg/5.2/var-name.html ↗

7.4.1 Commands

Commands can be embedded in running text or presented as part of a screen environment. In running text, use `<command/>` when referring to an actual command you would use on a command line:

```
To start LibreOffice from the command line, use
<command>loffice</command>.
```

Where options or subcommands belong with a command, include them within the element `<command/>` itself:

```
To start LibreOffice Writer from the command line, use
<command>loffice --writer</command>.
```

If options or subcommands stand for themselves in a text, wrap them in the element `<option/>`. Use markup for commands even inside `<screen/>` environments. To avoid spelling or capitalization errors, whenever possible, try commands before adding them to the documentation. See also [Section 7.4.5, “Prompts”](#).

7.4.2 File names

A file name is the name of a file on a local or network disk. Can contain a simple name or include a path or other elements specific to the operating system. See also [Section 6.8, “File and directory names”](#).

```
Find the log file <filename>configuration.xml</filename>
in the directory <filename>/etc/sysconfig</filename>.
```

To assign standard names to files and images in DocBook and AsciiDoc, follow the naming conventions at <https://github.com/SUSE/doc-modular/blob/main/templates/README.md> ↗.

7.4.3 Literals

Use `<literal/>` to mark up data taken literally from a computer system.

```
To create a comment, insert <literal>#</literal> characters.
```

7.4.4 Placeholders

To mark up text that readers need to replace, use the `<replaceable/>` element. Capitalize placeholder text in all contexts where this is not detrimental to content intelligibility. Do not use spaces within placeholders. Instead, use underscores (`_`).

```
To list the contents of a directory, execute  
<command>ls <replaceable>DIRECTORY</replaceable></command>.
```

7.4.5 Prompts

When documenting commands entered into Bash with a `<screen/>` element, always prefix them with a prompt marked up this way:

```
<prompt>&gt;&nbsp;</prompt><command>ls</command>
```

To avoid making prompts longer than necessary, do not include paths, user or host names, unless this is vital to understanding. The first restricted user must always be named *tux*. For more information about names of restricted users, see [Chapter 3, Names of example items](#).

Whenever you provide commands in a `<screen/>`, make it clear if the user needs regular or elevated privileges. Avoid using root prompts in your documentation by using the `sudo` command where applicable. If you do need a root prompt, always mark it up as follows:

```
<prompt role="root">#&nbsp;</prompt><command>yast</command>
```

When documenting prompts other than the one of Bash, use a custom prompt that is as generic as possible.

For consistency, it is helpful to create entities for the prompts used in your documentation. Doc Kit repository contains entities for *user*, *root* and *sudo* prompts. For more information, see [Section 9.3, "Entities"](#).

7.4.6 Screens

Screens are used to present:

- long commands and commands together with their output
- system output, such as system messages
- code or configuration examples

```
<screen><prompt>tux &gt; </prompt><command>ls </command>
bin  dev  lib      mnt      proc  sbin      suse  usr
boot etc  lib64     mounts  root  selinux   sys   var
data home lost+found opt      run   srv       tmp</screen>
```

- Use screens only where necessary for understanding the documentation. Present longer screens as examples. For more information, see [Section 7.7, “Examples”](#).
- Do not add empty lines at the beginning or end of screens. They can be cut away by the SUSE style sheets. However, most other style sheets do not have such functionality.
- Text in screens must not follow the indentation level of the XML around them: All indentation will be reproduced verbatim.
- Lines in a screen must be at most 80 characters long. If you are working in a structure with less available space, such as within a list or within a table, work with appropriate shorter line lengths.
- Avoid command output that contains dates, version numbers, or other version-specific information that frequently changes.
- To make long shell commands less unwieldy, split them into multiple lines at appropriate positions, such as after the end of an option. At the end of each line but the last, append a `\`. The character `\` informs the shell that the command invocation will continue after the end of the line. Splitting commands into lines can also be helpful for aligning callouts with the right option.
- You can combine multiple commands within a `<screen/>` element, if:
 - all commands are explained unambiguously before the `<screen/>` element and
 - all commands are strictly consecutive and none is optional.

In all other cases, do not combine multiple commands within one `<screen/>` element. Instead, use multiple `<screen/>` elements. For example, as part of a multi-step procedure.

If you show multiple commands within a single `<screen/>`, use prompts before each command and `<command/>` elements around commands. This effectively separates commands from each other and their output.

- To work with long output, especially tabular output, use either of the following strategies:
 - Remove or replace items that are irrelevant to your goal. For example, replace long file names with shorter ones or remove a table column and replace it with `[. . .]`.
 - Use a processing instruction at the beginning of the screen to decrease font size:

```
<?dbsuse-fo font-size="SIZEem"?>
```

Replace `SIZE` with a suitable value, such `0.7`. Choose a value between `0.55` and `1`. Values outside that range will lead to either unreadably small or unsuitably large text.

- To enable automatic syntax highlighting for programming languages or formats, add a `language` attribute for the respective language format. Valid language formats: `apache`, `bash`, `c++`, `css`, `diff`, `html`, `xml`, `http`, `ini`, `json`, `java`, `javascript`, `makefile`, `nginx`, `php`, `perl`, `python`, `ruby`, `sql`, `crush`, `dockerfile`, `lisp`, and `yaml`.

Note that syntax highlighting may not be supported in all target formats.

See also [Section 7.7, “Examples”](#), [Section 7.4.1, “Commands”](#), [Section 7.4.5, “Prompts”](#), and [Section 7.3, “Callouts”](#).

7.4.7 Variable names

To reference to names of variables, use the `<varname/>` element:

```
To select another display manager, start the YaST system configuration editor
and change the value of <varname>DISPLAYMANAGER</varname>.
```

7.5 Cross-references

Use the `<xref/>` element (read: “cross ref”) when referring to an appendix, chapter, example, figure, part, preface, section, table or question and answer set. The element referenced needs to have an `xml:id` attribute, an identifier. Create identifiers to reference from cross-references using the rules under [Section 7.12, “Identifiers”](#). Note that the `<xref/>` element only works when it links to documentation within the same set, for example, the same book or set of books.

Other types of references to resources are described in [Section 7.20, “References to other external resources”](#) and [Section 7.8, “External links”](#).

EXAMPLE 7.6: EXAMPLE OF A CROSS-REFERENCE (SOURCE)

```
<sect2 xml:id="sec-cross-reference">
  <title>Cross-references</title>
  <para>
    Use the <sgmltag class="emptytag">xref</sgmltag> element ...
  </para>
  ...
  <para>
    See <xref linkend="sec-cross-reference"/>.
  </para>
</sect2>
```

EXAMPLE 7.7: EXAMPLE OF A CROSS-REFERENCE (OUTPUT)

See [Section 7.5, “Cross-references”](#).

This example shows the default display of a cross-reference. To change it, use the `xrefstyle` attribute, either with `select:` or `template:`. Do not use custom “named” `xrefstyle` attributes that require support from the style sheets. For more info, refer to [DocBook XSL: The Complete Guide: Customizing cross-references \(http://www.sagehill.net/docbookxsl/CustomXrefs.html\)](http://www.sagehill.net/docbookxsl/CustomXrefs.html).

Keep in mind the following cases where listing cross-references is discouraged and must be avoided:

- Do not insert cross-references (`<xref linkend="...">`) into a `<title/>` element. The title must not be clickable, and a cross-reference in a title can create issues when linking to such a title in a different paragraph.
- Do not create references to paragraphs (`<para/>`) or other elements that have no title. An exception to this rule is the element `<step/>` to which you may create references. If a reference to an element without a title is vital to the document, use the attribute `xreflabel` to assign a title.

Do not prefix or suffix cross-references with text labels such as “appendix,” “chapter,” “table,” or “figure.” Such labels are generated automatically.

7.6 Emphasis

Where possible, indicate stress with language only. If that is not possible, use the `<emphasis/>` element to indicate stress.

Where added emphasis is needed, use the `role="bold"` attribute.

```
This will be displayed in <emphasis>italics</emphasis>. This
will be displayed in <emphasis role="bold">bold</emphasis>
```

7.7 Examples

Use examples to illustrate complex processes. The rules established in [Section 7.10.1, “Graphics”](#) also apply to examples.

Examples usually contain `<screen/>` elements. Additionally, there can be callouts and explanatory text.

Always give examples a title and an identifier.

For more information about screen environments, see [Section 7.4.6, “Screens”](#). For more information about displaying computer input and output, see [Section 7.4, “Command-line input and command-line output”](#). To annotate examples, use callouts. Callouts are described in [Section 7.3, “Callouts”](#).

EXAMPLE 7.8: EXAMPLE OF AN EXAMPLE

```
<example xml:id="ex-example">
  <title>Example of an example</title>
  <screen><prompt>tux &gt; </prompt><command>ps -xa</command>

5170 ?      S      0:00 kdeinit: khotkeys
5172 ?      S      0:02 kdeinit: kdesktop
5174 ?      S      0:04 kdeinit: kicker</screen>
</example>
```

TABLE 7.4: ELEMENTS RELATED TO `<example/>`

Element	Web link
<code><example/></code> . Formal block element containing a <code><title/></code> and a <code><screen/></code> or other elements such as lists or paragraphs.	https://tdg.docbook.org/tdg/5.2/example.html ↗
<code><screen/></code> . Verbatim block element for displaying text that readers might see on a computer terminal or in a text file.	https://tdg.docbook.org/tdg/5.2/screen.html ↗

7.8 External links

Information that is relevant within SUSE documentation is often available from other Web sites already. In such cases, choose between linking to these sites or including their content in edited form. Adhere to the following guidelines when selecting sites to link to:

- Link to credible sources, such as suse.com, upstream projects or developer sites. Avoid linking to direct competitors of SUSE. Do not link to obvious clickbait sites.
- Prefer larger, well-kept sites over blogs that may vanish overnight. If you need to link to smaller blogs, save an archive version of the site at <https://web.archive.org/>.
- Avoid linking to sites that feature controversial or political content.

In some cases, product managers will request avoiding all or selected external links to avoid issues for customers impacted by restrictive firewall rules.

Use the `<link/>` element to mark up URLs that can be opened with a Web browser, such as <https://www.example.org/>. Always add the correct protocol prefix (for example, `https://`), otherwise links will not work. If possible, use the secure protocol prefix (`https://`).

Never use `filename` for a link, as that would both disable the link checker and make the link unclickable. Avoid entering a text label between `<link/>` start and end tags. Instead, use a self-closing tag:

```
<link xlink:href="https://www.example.org/" />
```

Make URLs as short as possible before adding them to documentation. Many long URLs can be shortened by leaving away non-essential pieces, such as the `_utm` parameters used for marketing purposes. If a Web site provides a built-in URL shortener, use it.

Do not use third-party URL shorteners, such as bit.ly. Third-party URL shorteners have the following disadvantages:

- They obscure the destination a link points to.
- They introduce an extra element of uncertainty, as the shortening service may disappear or become unreliable in the future.
- The providers of these services usually run Web analytics that may introduce privacy issues.

Do not use `<link/>` to link to SUSE documentation outside of the current document set. Instead, use the appropriate entity for the book title. Always reference the book itself, as chapter names can change.

For consistency, do not use the article in front of the name of the referenced book or chapter. For example, “The general concept of Podman is described in *Containers and Podman*.”

Where possible, collect links in a “More information” section at the end of the chapter. This helps readers focus on your documentation instead of leading them immediately to other resources. This is described in [Section 4.1, “Books”](#).

To mark up multiple links, create an `<itemizedlist/>` around them. Do not use a list environment for a single link. If you need to present many 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. For more information about creating lists, see [Section 7.13.1, “Itemized lists”](#).

Where possible, provide translators with localized versions of links in the comments of the source file.

Other types of references to resources are described in [Section 7.5, “Cross-references”](#) and [Section 7.20, “References to other external resources”](#).

7.9 External links to SUSE documentation

The SUSE documentation is hosted under documentation.suse.com. This is the URL that must be provided in all documents. The abbreviated URLs such as doc.suse.com and docs.suse.com also work but must be avoided for SEO reasons.

Make sure to use complete URLs instead of entities for an easy copy and paste.

Most links in our documentation that goes to documentation.suse.com refer to a specific product and release. However, sometimes it makes sense to *not* include the SP or even the major release. These are so-called *SP-independent links*.

The following reasons give you an idea when to use them:

- Your linked information is independent from any SP
- You cannot or do not need to pick a specific SP
- You want to link to the most recent SP without checking which one it is
- You want to support SEO where the most prominent SP is more important than previous, older SPs

Make sure to follow this syntax:

```
documentation.suse.com/<PRODUCT>[/<MAJOR_RELEASE>]/<DEEP_LINK>
```

The placeholders mean the following:

- PRODUCT: the abbreviation of the product, like “sle” for SUSE Linux Enterprise, “sle-ha” for SUSE Linux Enterprise Server High Availability Extension, etc.
- MAJOR_RELEASE: an optional major release like 12 or 15. If you omit it, your link will be redirected to use the most recent release.
- DEEP_LINK: the link that points to a specific chapter or section within a book

Make sure you use html and not single-html. This is needed for SEO reasons.

For example, the link <https://documentation.suse.com/sle-ha/15-SP3/html/SLE-HA-all/article-installation.html#sec-ha-inst-quick-req> points to the “System requirements” for SUSE Linux Enterprise Server High Availability Extension. To turn this link into an SP-independent link, you need to identify the different components first:

- PRODUCT is sle-ha
- MAJOR_RELEASE is 15 (no SP mentioned)
- DEEP_LINK is /html/SLE-HA-all/article-installation.html#sec-ha-inst-quick-req

With the above parts, the redirection rules on our server allow expressing SP-independent links in several ways:

- <https://documentation.suse.com/sle-ha-15/html/SLE-HA-all/article-installation.html#sec-ha-inst-quick-req>
Redirects to the most recent SP for the major release 15
- <https://documentation.suse.com/sle-ha/html/SLE-HA-all/article-installation.html#sec-ha-inst-quick-req>
Redirects to the most recent major release, latest SP available



Note

If IDs have been changed between releases or SPs, this is what happens:

- If the hash part (everything after #) is not found, the browser will jump to the beginning of the file.
- If the file cannot be found (in our example, article-installation.html), the server will respond with a 404 error (file not found).

7.10 Figures

For figures within lists or procedures, use the `<informalfigure/>` element. In all other cases, use the `<figure/>` element. Always assign an `xml:id` attribute to `<figure/>` elements. Reference figures from the text by means of a cross-reference. For more information, see [Section 7.5, “Cross-references”](#).

All referenced image files must have a lowercase alphanumeric file name. When specifying figure names, follow the naming conventions at <https://github.com/SUSE/doc-modular/blob/main/templates/README.md>.

Provide an appropriate image width using the `width` attribute. For both figure types, formal and informal, always add a `<textobject role="description"/>` as in [Example 7.9, “Example of a figure”](#) to provide an alternative text for the HTML output.

EXAMPLE 7.9: EXAMPLE OF A FIGURE

```
<figure xml:id="fig-picture">
  <title>An interesting picture</title>
  <mediaobject>
    <imageobject role="fo">
      <imagedata fileref="picture.eps" width="80%" format="EPS"/>
    </imageobject>
    <imageobject role="html">
      <imagedata fileref="picture.png" width="80%" format="PNG"/>
    </imageobject>
    <textobject role="description">
      <phrase>Cat chasing Geeko</phrase>
    </textobject>
  </mediaobject>
</figure>
```

TABLE 7.5: ELEMENTS RELATED TO `<figure/>`

Element	Web link
<code><figure/></code> . Formal block element containing a <code><title/></code> and a <code><mediaobject/></code> .	https://tdg.docbook.org/tdg/5.2/figure.html ↗
<code><informalfigure/></code> . Informal block element containing a <code><mediaobject/></code> .	https://tdg.docbook.org/tdg/5.2/informalfigure.html ↗
<code><mediaobject/></code> . Block element containing one or more <code><imageobject/></code> elements. Place additional textual descriptions inside <code><textobject/></code> elements.	https://tdg.docbook.org/tdg/5.2/mediaobject.html ↗
<code><imageobject/></code> . Element containing <code><imagedata/></code> and meta information about the image.	https://tdg.docbook.org/tdg/5.2/imageobject.html ↗
<code><imagedata/></code> . Element that points to an external image file.	https://tdg.docbook.org/tdg/5.2/imagedata.html ↗
<code><textobject/></code> . Element containing textual description of a media object as a fallback option.	https://tdg.docbook.org/tdg/5.2/textobject.html ↗

7.10.1 Graphics

Keep graphics as simple as possible. Use as little text as possible. To allow for translation, reserve twice as much space for runs of text as the English version of it consumes.

7.10.2 Screenshots

Use screenshots to illustrate complex situations in which the user cannot easily follow the instructions otherwise.

- Be selective. Only illustrate steps in which meaningful user interactions are necessary. Do not create screenshots of progress bars or confirmation windows. Usually, it is unnecessary to create a screenshot of every step of an instruction.
- Always create screenshots illustrating the situation right before an action has been taken.
- Insert screenshots directly after the textual description of the action.
- Make sure screenshots focus on what they are supposed to illustrate. When documenting application windows, create a screenshot of the window only. When documenting Web applications, only reproduce the contents of the tab instead of the entire browser window.
- Do not scale screenshots using graphics software. Embed screenshots at their original resolution and use DocBook attributes to scale them appropriately.
- Avoid creating screenshots of windows higher or wider than 800 pixels at 96 pixels per inch. When creating screenshots of applications scaled for a higher pixel-per-inch count, apply a proportionally larger maximum window size.
- To ensure readability and consistency, scale screenshots with the `width` attribute. Choose the appropriate scaling from the following list:
 - Screenshots of the whole desktop should be scaled to 90–99% page width.
 - Screenshots of individual application windows should be scaled to 75–99% page width.
 - Small windows such as message boxes should be scaled to 50–60% page width.
- Create screenshots that are recognizable to readers. For example, create screenshots of KDE applications on a KDE desktop with the default KDE theme and disable toolbar modifications you have made.
- Use grayscale font antialiasing (default on SUSE operating systems). Subpixel font antialiasing (default on Windows and macOS operating systems) creates colored letter edges when zoomed or printed.
- Where applicable, follow the rules in *Chapter 3, Names of example items*.

- Avoid editing screenshots. To anonymize portions of a screenshot, pixelize it. To highlight parts of a screenshot, use rectangles or arrows. Never add callouts, text or freely drawn objects. Always select colors that provide a good contrast with their background.
- If possible, avoid screenshots with dates, version numbers, or other version-specific information that frequently changes.

7.11 Glossaries

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

Define infinitive forms of verbs and singular nouns. Do not start the definition with the term itself. Use lowercase for the term unless it is a proper noun.

Where applicable, group your terms with the `<glossdiv/>` tag for higher ordering.

To support automatic alphabetical ordering in localized versions, use `<glosslist/>` so that glossary items are sorted in alphabetical order by default.

Use cross-references in the following cases:

- To direct the reader to another glossary entry, use the `<glosssee/>` tag. This is helpful, for example, when linking acronyms with their written out forms.
- To provide the reader with additional information about related glossary entries, use `<glossseealso/>`.

The markup for a glossary entry is shown in *Example 7.10, "A typical example of a glossary"*.

EXAMPLE 7.10: A TYPICAL EXAMPLE OF A GLOSSARY

```
<glossary xmlns="http://docbook.org/ns/docbook" version="5.2">
  <title>Glossary</title>

  <glossentry xml:id="gt-docbook">
    <glossterm>DocBook</glossterm>
    <glossdef><para>...</para></glossdef>
  </glossentry>

  <glossentry xml:id="gt-svg">
    <glossterm>SVG</glossterm>
    <glossdef><para>...</para></glossdef>
```

```

</glossentry>

<glossentry xml:id="gt-extensible-markup-language">
  <glossterm>Extensible Markup Language</glossterm>
  <glossdef>
    <para> A markup language that defines a set of rules for encoding
      documents in a format that is both human-readable and
      machine-readable. </para>
    <glossseealso linkend="gt-docbook"/>
    <glossseealso linkend="gt-svg">SVG (Scalable Vector Graphics)</glossseealso>
  </glossdef>
</glossentry>

<glossentry xml:id="gt-xml">
  <glossterm>XML</glossterm>
  <glosssee linkend="gt-extensible-markup-language" />
</glossentry>

</glossary>

```

7.12 Identifiers

- Always use an `xml:id` attribute in parts, chapters, appendixes, sections, figures, glossaries and examples. Identifiers can be used in other elements as well, for example, block elements, such as tables and procedures.
- In identifiers, only use lowercase basic Latin alphabetic and numeric characters and `-` (hyphen). Follow the regex pattern `[\-0-9a-zA-Z]+`. Do not use `_` and `.` characters, as they may hurt search engine optimization.
- Identifiers can consist of multiple parts. Join these parts with a `-` (hyphen).
 1. **Prefix.** Signifies the type of XML element. Prefixes aid writers in creating logically named identifiers for elements. Use identifiers in accordance with *Table 7.6, "Abbreviations for different elements in an `xml:id` attribute"*.

Do not add prefixes to identifiers for XML elements that are used to determine the name of HTML pages (such as section and chapter elements). Such exposure of identifiers can hurt SEO.

2. **Chapter title label.** Shortened version of the title of the parent chapter or parent chapter-level element (preface, appendix, etc.). Do not add a chapter title label to chapters and chapter-level elements themselves. Do not add chapter title identifiers within articles.

3. **Element title label.** Shortened version of name of the title of the element itself.

EXAMPLE 7.11: EXAMPLES OF IDENTIFIERS

```
xml:id="pro-install-sles"  
xml:id="install-yast"  
xml:id="tab-install-source"
```

- Use short, memorable, English terms or phrases as title labels. Favor longer terms over non-obvious abbreviations. Always use the singular of nouns and the infinitive of verbs. For example, a section about installing with YaST could be called `install-yast`. A figure in that section showing language selection could use the identifier `fig-in-stall-yast-language`.
- Keep in mind that section and chapter identifiers are used in the online documentation URLs. Choosing understandable keywords helps readers to understand what the page is about and also improves the search engine ranking.

Do not rework identifiers in existing documentation, instead apply these rules to newly created documentation only.

TABLE 7.6: ABBREVIATIONS FOR DIFFERENT ELEMENTS IN AN `xml:id` ATTRIBUTE

Element	Prefix
<code><appendix/></code>	<i>No prefix</i>
<code><book/></code>	<i>No prefix</i>
<code><co/></code>	<code>co</code>
<code><chapter/></code>	<i>No prefix</i>
<code><example/></code>	<code>ex</code>
<code><figure/></code>	<code>fig</code>

Element	Prefix
<u><glossary/></u>	<i>No prefix</i>
<u><glossterm/></u>	<u>gl</u>
<u><itemizedlist/></u>	<u>il</u> ^a
<u><listitem/></u>	<u>li</u>
<u><indexterm/></u>	<u>idx</u> ^b
<u><orderedlist/></u>	<u>ol</u> ^a
<u><part/></u>	<i>No prefix</i>
<u><preface/></u>	<i>No prefix</i>
<u><procedure/></u>	<u>pro</u>
<u><qandaset/></u> , <u><qandadiv/></u> , <u><qandaentry/></u>	<u>qa</u>
<u><sect1/></u> , <u><sect2/></u> , etc., <u><section/></u>	<i>No prefix</i>
<u><set/></u>	<i>No prefix</i>
<u><step/></u>	<u>st</u>
<u><table/></u>	<u>tab</u>
<u><topic/></u>	<i>No prefix</i>
<u><variablelist/></u>	<u>vl</u>
<u><varlistentry/></u>	<u>vle</u>

a Only add an `xml:id` attribute when the list has a title element

b Only add when creating index ranges

7.13 Lists

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

- Itemized lists (`<itemizedlist/>`). Also known as bullet lists or unordered lists.
- Ordered lists (`<orderedlist/>`). Also known as numbered lists.
- Variable lists (`<variablelist/>`). Also known as definition lists or description lists.
- Procedures (`<procedure/>`). Also known as step-by-step instructions or step lists. Described in [Section 7.16, "Procedures"](#).

Follow these rules when creating or editing lists:

- Always introduce a list in the text. If needed for reference or better coordination with the related text, add a title and an `xml:id` attribute.
- A list must contain at least two items. If items are few, short and simple in structure, consider incorporating them in the flowing text instead of creating a list.
- If all list items are nouns only, do not capitalize their first letter. Use sentence-style capitalization for list items that are full sentences and for terms in descriptive lists.
- Use a period after every list item that is a sentence. Do not use a period after the items that are not complete sentences. Use either all full sentences in your bullet lists or all fragments. Avoid a mix.
Do not use commas and semicolons to end punctuation.
- Wherever possible, use parallel phrasing and grammatical construction between list items. This provides a pattern that makes it easier to follow the text.
- Lists are visually distinct and can break up text flow. Do not overuse them.

Never nest more than three lists within each other. Instead, restructure the information using a combination of lists and running texts.

To be able to reference untitled lists, use the `xreflabel` attribute. For more information, see [Section 7.5, "Cross-references"](#).

TABLE 7.7: ELEMENTS RELATED TO LISTS

Element	Web link
<code><itemizedlist/></code> . Block element for an unordered list. Contains multiple <code><listitem/></code> elements.	https://tdg.docbook.org/tdg/5.2/itemizedlist.html ↗
<code><orderedlist/></code> . Block element for a numbered list. Contains multiple <code><listitem/></code> elements.	https://tdg.docbook.org/tdg/5.2/orderedlist.html ↗
<code><variablelist/></code> . Block element for a descriptive list. Contains multiple <code><varlistentry/></code> elements.	https://tdg.docbook.org/tdg/5.2/variablelist.html ↗
<code><varlistentry/></code> . Element within a <code><variablelist/></code> that associates a <code><term/></code> and a <code><listitem/></code> .	https://tdg.docbook.org/tdg/5.2/varlistentry.html ↗
<code><term/></code> . Element whose content serves as the title of an element of a <code><variablelist/></code> .	https://tdg.docbook.org/tdg/5.2/term.html ↗
<code><listitem/></code> . A single list element. To add text to this item, first add a <code><para/></code> element.	https://tdg.docbook.org/tdg/5.2/listitem.html ↗

7.13.1 Itemized lists

Use itemized lists whenever the order of list items is irrelevant. They are often used to provide an overview of information or to introduce or summarize information.

EXAMPLE 7.12: EXAMPLE OF AN ITEMIZED LIST (SOURCE)

```
<para>
  The following operating systems are supported:
</para>
<itemizedlist>
  <listitem>
    <para>
```

```
Linux, Kernel 2.4 and newer
</para>
</listitem>
<listitem>
  <para>
    FreeBSD 7 and newer
  </para>
</listitem>
</itemizedlist>
```

EXAMPLE 7.13: EXAMPLE OF AN ITEMIZED LIST (OUTPUT)

The following operating systems are supported:

- Linux, Kernel 2.4 and newer
- FreeBSD 7 and newer

7.13.2 Ordered lists

Use ordered lists when items have a strict order, hierarchy or importance. Do not use ordered lists to describe sequential user actions (step-by-step instructions). For sequential user actions, use a procedure, as described in [Section 7.16, “Procedures”](#). If order is not relevant, use an itemized list or a variable list.

EXAMPLE 7.14: EXAMPLE OF AN ORDERED LIST (SOURCE)

```
<para>
  Before installing, make sure of the following:
</para>
<orderedlist>
  <listitem>
    <para>
      The network connection of the computer is configured properly.
    </para>
  </listitem>
  <listitem>
    <para>
      The latest security updates are installed.
      If you are in doubt, run an online update.
    </para>
  </listitem>
</orderedlist>
```

EXAMPLE 7.15: EXAMPLE OF AN ORDERED LIST (OUTPUT)

Before installing, make sure of the following:

1. The network connection of the computer is configured properly.
2. The latest security updates are installed. If you are in doubt, run an online update.

7.13.3 Variable lists

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

Use sentence-style capitalization for both the term and the list item.

To reference the list, assign it a `xml:id` attribute and add a title. Individual list items may be referenced by assigning an `xml:id`. The entry is then identified by the value of `xml:id` and referenced by the term.

EXAMPLE 7.16: EXAMPLE OF A VARIABLE LIST (SOURCE)

```
<para>
  This book consists of several parts:
</para>
<variablelist>
  <varlistentry>
    <term>Installation</term>
    <listitem>
      <para>
        Learn about the installation and initial configuration of a Linux system.
      </para>
    </listitem>
  </varlistentry>
  <varlistentry>
    <term>System</term>
    <listitem>
      <para>
        Get a basic understanding of the system components.
      </para>
    </listitem>
  </varlistentry>
</variablelist>
```

EXAMPLE 7.17: EXAMPLE OF A VARIABLE LIST (OUTPUT)

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.

7.14 Keys and key combinations

Capitalize all keys as printed on a standard keyboard. Capitalize all letter keys. To refer to a capitalized character, use `Shift - Z`, for example. Introduce this convention by means of the “Typographical Conventions” section of the introduction.

To mark up key combinations, use `<keycombo/>` as a wrapper for multiple `<keycap/>` elements. Separators between `<keycap/>` elements are then created automatically.

If a key is listed in *Table 7.8, “Elements related to `<keycap/>`”*, use the `function` attribute with the appropriate value. When using the `function` attribute, make the tag self-closing—DocBook's language files will insert key names automatically. This simplifies both your work and the work of translators.

For more information about creating cross-references, see *Section 7.5, “Cross-references”*.

EXAMPLE 7.18: EXAMPLE OF A KEY

To create a screenshot, press `<keycap>Print Screen</keycap>`.

EXAMPLE 7.19: EXAMPLE OF A KEYBOARD COMBINATION

To save a file, press
`<keycombo><keycap function="control"/><keycap>S</keycap></keycombo>`.

TABLE 7.8: ELEMENTS RELATED TO `<keycap/>`

Element	Web link
<code><keycombo/></code> . Inline element containing multiple <code><keycap/></code> elements that together make up a key combination.	https://tdg.docbook.org/tdg/5.2/key-combo.html ↗
<code><keycap/></code> . Inline element to mark up a single key. Contains either the key labels text inside it or is self-closing and has a <code>func-</code>	https://tdg.docbook.org/tdg/5.2/keycap.html ↗

Element	Web link
<p>tion attribute with one of the following values:</p> <ul style="list-style-type: none"> • alt • backspace • command • control • delete • down • end • enter • escape • home • insert • left • meta (also known as <code>Win</code> , <code>Windows</code> or <code>Super</code>) • option (macOS only) • pagedown • pageup • right • shift • space 	

Element	Web link
<ul style="list-style-type: none"> • tab • up 	

7.15 Outline levels and sectioning

Create sections using the [<sect1/>](#), [<sect2/>](#) and [<sect3/>](#) elements. Avoid outlines that require [<sect4/>](#) and [<sect5/>](#) elements. Instead, create a flatter structure in which more elements are visible at a glance.

Provide at least one paragraph of introductory information directly within each section.

Do not create lone subsections. A lone subsection is a section that is the only subsection of its parent section.

For more information about writing headlines, see [Section 6.9, “Headings”](#).

7.16 Procedures

Use procedures to describe sequential tasks. A procedure consists of the following elements and attributes:

- An [xml:id](#) attribute.
- A title.
- An introductory phrase establishing the purpose of the procedure. If the procedure is otherwise the only element in its section, place the introductory phrase before the procedure.
- If there are preconditions or prerequisites, add them as a second paragraph after the introduction.
- Short, simple steps and, if necessary, substeps describing the actions to be performed. See also [Section 6.17, “Sentence structure”](#).

To link alternative actions inside the same substep element, use “or.” Apply a [performance=optional](#) attribute to optional steps.

Steps may contain a link to an explanatory subsection providing further details on the step.

EXAMPLE 7.20: EXAMPLE OF A PROCEDURE (SOURCE)

```
<procedure xml:id="pro-procedure">
  <title>Example of a procedure</title>
  <para>
    To add a new user to the system, perform the following steps:
  </para>
  <step>
    <para>
      In the <phrase role="productname">YaST</phrase> window,
      click <guimenu>User and group management</guimenu>.
    </para>
  </step>
  <step>
    <para>
      To open the <guimenu>Add a new user</guimenu> dialog, click
      <guimenu>Add</guimenu>.
    </para>
  </step>
  <step>
    <para>
      Type in the user name and click <guimenu>Create</guimenu>.
    </para>
  </step>
</procedure>
```

PROCEDURE 7.1: EXAMPLE OF A PROCEDURE (OUTPUT)

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

1. In the YaST window, click *User and group management*.
2. To open the *Add a new user* dialog, click *Add*.
3. Type in the user name and click *Create*.

TABLE 7.9: ELEMENTS RELATED TO `<procedure/>`

Element	Web link
<code><procedure/></code> . Block element containing a <code><title/></code> (optional) and multiple <code><step/></code> elements.	https://tdg.docbook.org/tdg/5.2/procedure.html ↗
<code><step/></code> . Element signifying a single unit of action. Usually contains a <code><para/></code> element, but can also house a <code><substeps/></code> element.	https://tdg.docbook.org/tdg/5.2/step.html ↗

Element	Web link
<code><substeps/></code> . Element containing multiple, subordinate <code><step/></code> elements.	https://tdg.docbook.org/tdg/5.2/substeps.html 

7.17 Products

Always use the preferred product name instead of, for example, an acronym. When referring to a product, add a `<phrase role="productname"/>` element around it. This will not result in a visual change but disables hyphenation:

```
<phrase role="productname">LibreOffice</phrase> is an office suite.
```

7.18 Profiling

Profiling is convenient for the creation of consistent documentation across different products or product lines. This is especially beneficial when similar products share a considerable amount of features, with only a few differences. Instead of maintaining separate documentation for each product, you can share most of the XML source code and only vary text snippets where necessary. In DocBook XML files, you can mark some elements as conditional by using profiling attributes. Specify which conditions apply to the output when processing the files to generate output. The style sheets will then include or exclude the marked text, according to the conditions.

Profiling allows you to keep both common and product-specific content in one XML file and select at production time which information to include in the output.

If you need to use profiling in your writing, adhere to the following guidelines:

- Identify different variants that you want to apply to the general piece of text and assign clear and short identifiers to them, sticking to lowercase. These identifiers act as “aliases” for longer products or deliverables. If you want to apply more than one identifier to an element, separate them with a semicolon.
- Select one or more profiling attributes and add them to the text snippets that are conditional. The tagged snippets will only be included in the output if all required conditions are fulfilled. In most cases, the attribute to use is `os`. For different processor architectures, use `arch`. The general-purpose attribute is `condition`.

- Mark the variants in your text with the relevant identifiers. Any content that is valid for all conditions does not need any profiling attributes. The respective content will always be included in the output formats generated from the XML sources.
- Create a different DC file for each variant. Add the respective profiling variable and its value to the DC file.
- DocBook allows you to use multiple profiling attributes to handle more complex scenarios. For example, if you have conditions for architecture (`arch`) and operating system (`os`), you can create different versions of a document for different combinations of these conditions, as shown in *Example 7.23, “Multiple profiling with attributes `os` and `arch`”*.

EXAMPLE 7.21: SINGLE PROFILING WITH THE ATTRIBUTE `os`

```
<?xml version="1.0" encoding="UTF-8"?>
[...]
<phrase os="sles;sled">Note that the official update repository is only
available after registering your SUSE Linux Enterprise Server installation.</phrase>
```

EXAMPLE 7.22: DC FILE WITH PROFILING FOR SLES

```
MAIN="MAIN.SLEDS.xml"
ROOTID=book-administration

## Profiling
PROFOS="sles"
...
```

EXAMPLE 7.23: MULTIPLE PROFILING WITH ATTRIBUTES `os` AND `arch`

```
<?xml version="1.0" encoding="UTF-8"?>
[...]
Entire hard disks are listed as devices without numbers, such as
<filename>/dev/sda</filename><phrase arch="zseries;aarch64" os="sles;sled;slemicro"></
phrase>.
```

7.19 Questions and answers

Use questions-and-answers sections to present information about troubleshooting or commonly asked questions about a product. Never use questions-and-answers sections to explain trivia, such as how a product got its name. Keep your audience in mind. See also *Chapter 1, Writing technical documentation*.

Questions must always end in a ?. Where explanations longer than three paragraphs are necessary for an answer, add a cross-reference to an explanation outside of the questions-and-answers section. See also [Section 7.5, “Cross-references”](#).

When a questions-and-answers section contains over 10 questions and there are clear topical divisions, add `<qandadiv/>` elements to further structure the section.

EXAMPLE 7.24: [EXAMPLE OF A QUESTIONS-AND-ANSWERS SECTION \(SOURCE\)](#)

```
<qandaset>
  <title>Example of a questions-and-answers section</title>
  <qandaentry>
    <question>
      <para>
        How can I check if the product was correctly installed?
      </para>
    </question>
    <answer>
      <para>
        Open the log file.
        Look for entries starting with <literal>Failed</literal>.
      </para>
    </answer>
  </qandaentry>
  <qandaentry>
    <question>
      <para>
        Why does the error <literal>Not enough disk space</literal> occur
        during installation?
      </para>
    </question>
    <answer>
      <para>
        There are less than 4 GB of space available on the selected partition.
      </para>
    </answer>
  </qandaentry>
</qandaset>
```

Example of a questions-and-answers section (output)

1. *How can I check if the product was correctly installed?*
Open the log file. Look for entries starting with Failed.

2. *Why does the error Not enough disk space occur during installation?*

There are less than 4 GB of space available on the selected partition.

TABLE 7.10: ELEMENTS RELATED TO `<qandaset/>`

Element	Web link
<code><qandaset/></code> . Block element containing a <code><title/></code> (optional) and multiple <code><qandaentry/></code> or <code><qandadiv/></code> elements.	https://tdg.docbook.org/tdg/5.2/qandaset.html ↗
<code><qandadiv/></code> . Block element containing a <code><title/></code> and multiple <code><qandaentry/></code> elements. Used to structure a <code><qandaset/></code> into smaller topical subsections.	https://tdg.docbook.org/tdg/5.2/qandadiv.html ↗
<code><qandaentry/></code> . Block element used to associate a <code><question/></code> with an <code><answer/></code> .	https://tdg.docbook.org/tdg/5.2/qandaentry.html ↗
<code><question/></code> . Block element containing the question. Use a single <code><para/></code> element inside.	https://tdg.docbook.org/tdg/5.2/question.html ↗
<code><answer/></code> . Block element containing the answer. Use <code><para/></code> elements inside.	https://tdg.docbook.org/tdg/5.2/answer.html ↗

7.20 References to other external resources

To reference file names, use the `<filename/>` element. To reference e-mail addresses, use the `<email/>` element. In either case, do not include a protocol prefix, that is `file://` or `mailto:`, respectively. See also *Section 7.4.2, “File names”*.

Reference man pages and info pages in this format:

- “the man page of `command`”
- “the info page of `command`”

In a situation where the category of the page is needed, append the category in parentheses. For example, use “(man 9 `command`).”

To learn more about subcommands, see the man page of

```
<command>command</command>.
```

Insert references to external (non-SUSE) physical books in the format “Title by Author (ISBN #00000000).” Inclusion of the ISBN is optional. Place the title in a `<citetitle/>` element. For example:

```
<citetitle>Lorem Ipsum</citetitle> by Dolores S. Amet  
(ISBN 0-246-52044-7) is a useful guide.
```

Note that `<citetitle/>` is not translated.

As an author, where possible, provide language-specific references to translators in XML comments (see also [Section 9.2, “XML comments”](#)). As a translator, look for corresponding language-specific resources where none have been provided. For URLs, provide only the language-specific version of a site. Use the English version as a fallback. For books, provide the title of the translated version along with the original title if such a translation exists.

Other types of references to resources are described in [Section 7.5, “Cross-references”](#) and [Section 7.8, “External links”](#).

7.21 Tables

Use tables to present many similar facts. Tables are easy to scan and compare. Always keep tables simple enough to not require long explanations even for readers unfamiliar with the topic.

A table always has a title and should have an `xml:id` attribute.

Typical use cases of tables include:

Lookup tables for specific information

Whenever users need to check for data that applies to them, create lookup tables. For example, use a table to sum up system requirements.

Matrix tables

Whenever users need to quickly check whether a specific combination of options works or not. For example, use a matrix table to visualize supported combinations or update options.

As there are use cases for tables, there are cases when *not* to use tables:

- Value and description pairs are better handled by means of a descriptive list.
- Wordy explanations or descriptions should not be used in tables.
- Complex layout constructs (screens, code) should not be used in tables.

Some general style tips for tables:

Structure the table to have more rows than columns

Readers prefer to skim for information that is aligned vertically. When designing tables, consider swapping rows and columns to provide a consistent user experience.

Narrow down columns

Compress the data in your table as much as you possibly can. Table cells with less content are more easily parsed by readers. Consider using icons, adding repeating words into column titles, or using shorter number formats.

Use colors to lead the reader's eyes

Use colors to make the information stand out. If you just want to highlight small bits, use bright colors. If you need to color entire cells, use pastels. This option should only be used with matrix type tables.

Use striped rows

Color every second row in light gray to make sure readers do not accidentally slip between lines. This should be the default for long tables. Do not use striped rows in matrix type tables with additional cell background coloring. You may also consider disabling striped rows for short tables to not confuse readers.

EXAMPLE 7.25: [EXAMPLE OF A TABLE \(SOURCE\)](#)

```
<informaltable>
<tgroup cols="2">
  <thead>
    <row>
      <entry>File System</entry>
      <entry>Maximum File Size</entry>
    </row>
  </thead>
  <tbody>
    <row>
      <entry>Ext2 (1 kB block size)</entry>
      <entry>16 GB</entry>
    </row>
    <row>
      <entry>Ext2 (2 kB block size)</entry>
      <entry>256 GB</entry>
    </row>
  </tbody>
</tgroup>
</informaltable>
```

EXAMPLE 7.26: EXAMPLE OF A TABLE (OUTPUT)

File System	Maximum File Size
Ext2 (1 kB block size)	16 GB
Ext2 (2 kB block size)	256 GB

TABLE 7.11: ELEMENTS RELATED TO `<table/>`

Element	Web link
<code><table/></code> . Formal block element that contains a <code><title/></code> and a <code><tgroup/></code> element.	https://tdg.docbook.org/tdg/5.2/cals.table.html ↗
<code><informaltable/></code> . Informal block element that contains a <code><tgroup/></code> element.	https://tdg.docbook.org/tdg/5.2/cals.informaltable.html ↗
<code><tgroup/></code> . Wrapper for the content of a table. Can contain one or more <code><colspec/></code> and one <code><thead/></code> . Contains a <code><tbody/></code>	https://tdg.docbook.org/tdg/5.2/tgroup.html ↗
<code><colspec/></code> . Element to define common properties for a column.	https://tdg.docbook.org/tdg/5.2/colspec.html ↗
<code><thead/></code> . Element to mark up a table head. Contains a <code><row/></code> element.	https://tdg.docbook.org/tdg/5.2/cals.thead.html ↗
<code><tbody/></code> . Element to mark up the table body. Contains multiple <code><row/></code> elements.	https://tdg.docbook.org/tdg/5.2/cals.tbody.html ↗
<code><row/></code> . Element to mark up a table row. Contains multiple <code><entry/></code> elements.	https://tdg.docbook.org/tdg/5.2/row.html ↗
<code><entry/></code> . Element to mark up a table cell.	https://tdg.docbook.org/tdg/5.2/entry.html ↗

7.22 User interface items

To mark up single user interface items, use `<guimenu/>`. To mark up nested menu structures, use `<menuchoice/>` as a wrapper for multiple `<guimenu/>` elements. Separators between `<guimenu/>` elements are then created automatically.

For more information about language aspects, see [Section 6.22, “User interface items”](#).

EXAMPLE 7.27: EXAMPLE OF A SINGLE USER INTERFACE ITEM

To open a file, click `<guimenu>Open</guimenu>`.

EXAMPLE 7.28: EXAMPLE OF NESTED USER INTERFACE ITEMS

To save a file, use
`<menuchoice><guimenu>File</guimenu><guimenu>Save</guimenu></menuchoice>`.

TABLE 7.12: ELEMENTS RELATED TO `<guimenu/>`


Element	Web link
<code><menuchoice/></code> . Inline element containing multiple <code><guimenu/></code> elements that together form a nested menu structure.	https://tdg.docbook.org/tdg/5.2/menu-choice.html ↗
<code><guimenu/></code> . Inline element to mark up a single user interface item.	https://tdg.docbook.org/tdg/5.2/guimenu.html ↗

8 DocBook tags

The SUSE and openSUSE user documentation is written with DocBook and uses a Relax NG schema that defines a restricted set of DocBook tags.

The following tables list and describe all DocBook tags used for writing most of the SUSE user documentation. They also show which tags are translated and which tags will be blocked for translation, which means that those tags stay in English. The tables are divided into four categories, listing the elements of each respective category.

With GeekoDoc 2, it is also possible to use the Internationalization Tag Set (ITS) to explicitly indicate whether a tag should be translated or not. For more information on ITS tags, see [Section 8.2, “Using ITS tags”](#).

For more information on GeekoDoc, see <https://opensuse.github.io/geekodoc/> 

For more information on all tags available in DocBook, see <https://tdg.docbook.org/tdg/5.2/> 

8.1 Using DocBook tags

8.1.1 Meta elements

All the elements at the section level and above, and many other elements, include a wrapper for meta information about the content. The meta information wrapper is designed to contain bibliographic information about the content (author, title, publisher, etc.) as well as other meta information such as revision histories, keyword sets and index terms.

Element name	Description	Translation
abstract	Use an abstract to summarize the information provided in a book, article, or set in five or fewer sentences.	Yes
author personame firstname surname	Name of an individual author	No
authorgroup author personame firstname	Wrapper for author information when a document has multiple authors or collaborators	No

Element name	Description	Translation
surname		
date	Date of publication or revision of a document	No
info dm:docmanager dm:bugtracker dm:url dm:component dm:product dm:assignee dm:version dm:translation dm:editurl	<u><info></u> : Wrapper to contain bibliographic information about the content and other meta info. <u><dm:PLACEHOLDER></u> : SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.	No
indexterm primary secondary	To create an index	Yes Legacy element, do not use for new documentation
keywordset keyword	A set of keywords describing the content of a document	No For TM parser settings, set to Yes.
legalnotice		Yes
productname	The formal name of a product	No
productnumber	A number assigned to a product	No
remark	A remark (or comment) intended for presentation in a draft manuscript	No
subtitle	The subtitle of a document. Often used for SUSE Best Practices guides	Yes
title	The text of the title for a section of a document or for a formal block-level element	Yes

Element name	Description	Translation
	For document titles, such as book, article and set titles, use title-style capitalization. Apply sentence-style capitalization to all running text and all types of headings and titles that are part of the document content.	
titleabbrev	The abbreviation of a title. For example, it can be used to show shorter titles in the table of contents.	Yes

8.1.2 Structure elements



Note

Structure elements are needed for sectioning and structuring your document, such as defining chapters, appendix, etc.

Most structure elements do not need translations. The child elements like <title> are translated.

Element name	Description	Translation
article	An article	Yes
appendix	An appendix in a book or article	No For TM parser settings, set to <i>Yes</i> .
bridgehead	A free-floating heading. Use sparingly and work with sections instead.	Yes
book	A book	No For TM parser settings, set to <i>Yes</i> .

Element name	Description	Translation
chapter	A chapter, as of a book	No For TM parser settings, set to <i>Yes</i> .
formalpara	A paragraph with the title rendered as a run-in head. Use to create more compact lists, for example: <u>who</u> . Lists currently logged in users. <u>w</u> . Shows who is logged in and what they are doing.	Yes
section	A recursive section, unnumbered	No For TM parser settings, set to <i>Yes</i> .
sect1 sect2 sect3 ...	Numbered sections that must be properly nested	No For TM parser settings, set to <i>Yes</i> .
set	A collection of books	Yes
para	A paragraph	Yes
part	A division in a book	No For TM parser settings, set to <i>Yes</i> .
preface	Introductory matter preceding the first chapter of a book	No For TM parser settings, set to <i>Yes</i> .

Element name	Description	Translation
qandaset qandaentry question answer	Used for FAQs	Yes

8.1.3 Block elements

The block elements occur immediately below the component and sectioning elements. These are the (roughly) paragraph-level elements in DocBook. They can be divided into a number of categories: lists, admonitions, line-specific environments, synopses of several sorts, tables, figures, examples, and a dozen or more miscellaneous elements.

Element name	Description	Translation
calloutlist callout para	List of callouts and their descriptions. A called out description of a marked area.	No For TM parser settings, set to <i>Yes</i> . Child element <u><para></u> is translated.
example	An example	Yes
figure title mediaobject imageobject imagedata	Figure with title	No For TM parser settings, set to <i>Yes</i> . Child element <u><title></u> is translated.
glossary glossdef glossdiv glossentry	A glossary	Yes

Element name	Description	Translation
glossterm		
important title	Use this elements to give vital information.	Yes
informalfigure	Figure without a title	No
informaltable	Table without a title	Yes
itemizedlist listitem	Unordered, bulleted list	Yes
note title para	A message set off from the text. Use this element to highlight software version differences.	Yes
orderedlist listitem	Numbered list with attributes to control the type of enumeration	Yes
procedure step substeps stepalternatives	<u><procedure></u> : A list of operations to be performed in a well-defined sequence <u><step></u> : A unit of action in a procedure	Yes
programlisting	A literal listing of a program or a part of a program. Use for program source or source fragment listings. Formatted as a displayed block. Similar to <u><screen></u> .	No
qandset qandaentry question answer	List of questions and answers, used for FAQs	Yes
screen	Text that a user sees on a computer screen	No
simplelist member	Undecorated list of single words or short phrases	Yes
table title	Formal table with title	No

Element name	Description	Translation
tgroup colspec thead row entry tbody row entry		For TM parser settings, set to <i>Yes</i> . Except for child elements <code><title></code> and <code><entry></code>
textobject	Explanatory text for images to aid visually impaired users and show when the image cannot be loaded because of an error	Yes
tip title para	To introduce guidelines or give tips	Yes
variablelist varlistentry term	List of terms and definitions or descriptions	Yes
warning title para	To warn of security issues, potential loss of data, damage to hardware, or physical hazards. Warnings must always precede the action to which they apply.	Yes

8.1.4 Inline elements

Inline elements are used to mark up running text. In published documents, inline elements often cause a font change or other small change, but they do not cause line or paragraph breaks.

Element name	Description	Translation
citetitle	To reference names of printed books. To refer to any of our guides, use <code><xref/></code> or <code><link/></code> .	No
co	Part of calloutlist. The location of a callout embedded in text.	No
code	An inline code fragment	No

Element name	Description	Translation
command	A software command	No
constant	A programming or system constant	No
email	An e-mail address	No
emphasis	Emphasized text	Yes
envvar	A software environment variable	No
filename	Name of a file or path as well as directories, printers or flash drives	No
function	Name of a function or subroutine, as in a programming language	No
guimenu	To mark up all GUI elements like button, label and menu	Yes Depends on whether UI is translated or not. If not, use ITS tag.
indexterm primary secondary	Index marker	Yes Legacy element, do not use for new documentation
inlinemediaobject	An inline media object (video, audio, image...)	No
keycap	The text printed on a key on a keyboard For function keys, always use with function, for example: <keycap function="control"> </keycap>	Yes
keycombo keycap	A combination of input actions on a keyboard	Yes

Element name	Description	Translation
link	A hypertext link. Use the secure protocol prefix (https://), if possible.	No
literal	To mark up data taken literally from a computer system	No
menuchoice	A selection or series of selections from a menu	Yes
option	An option for a software command	No
package	Name of a software or application package. Replaces <code><systemitem class="resource"></code> .	No
parameter	A value or a symbolic reference to a value	No
phrase	A span of text	Yes
productname	The formal name of a product	No
prompt	A character or string indicating the start of an input field in a computer display	No
quote	To quote from sources, such as books. In all other cases, do not use quotation marks.	Yes
replaceable	Content that may or must be replaced by the user. Capitalize placeholder text in all contexts where this is not detrimental to content intelligibility. Do not use spaces within placeholders, use underscores instead.	No
systemitem	A system-related item or term	No
subscript	A subscript. For example: H ₂ O	Yes
superscript	A superscript. For example: X ²	Yes

Element name	Description	Translation
tag	A component of XML (or SGML) markup	No
trademark	To mark a trademark with TM™	No
uri	For links that should not be clickable	No
varname	The name of a variable	No
xi:include	To construct documents from different files	No
xref	A cross-reference to another part of the document	No

8.2 Using ITS tags

With GeekoDoc 2, it is possible to use the Internationalization Tag Set (ITS) tags to explicitly indicate that a tag should be translated or not.

For this, you need to have the ITS namespace enabled in the XML file. Make sure the following line is added to the root element: `xmlns:its="http://www.w3.org/2005/11/its"`.

For example: You have used the tag `<guimenu>` which is usually translated but in this particular case it should not be translated.

To mark the tag correctly, use the ITS tag as shown here:

```
Click <guimenu its:translate="no">Save</guimenu>.
```

9 Managing documents

This section provides an overview over features of XML you can use to manage documents.

9.1 Remarks

Use remarks for editorial comments. Remarks can be placed within, before, or after a para but must always be within a section element. When creating output, remarks can be made visible in the output and thus help within the editorial process. When creating the final output, deactivate remarks.

Start remarks with your user name and the current date, then add a colon (:) and finally your actual remark. To comment on someone else's remark, add a new remark directly below it. Delete remarks when the corresponding issue is resolved.

```
<remark>tux (2013-10-13): could not find the option for foo</remark>
<remark>geeko (2013-11-02): see /usr/share/doc/foo.html</remark>
```

You can add a `role` attribute with one of the following values to show the type of the remark:

- `structure`. Use this type of remark to suggest changes to the text or XML structure.
- `language`. Use this type of remark to suggest language improvements.
- `needinfo`. Use this type of remark to mark sections where you need input from others, such as developers.
- `trans`. Use this type of remark to give hints to translators.

9.2 XML comments

XML comments can be used for temporarily disabling portions of text. Another use of XML comments is to create more permanent internal comments or to mark up changes made for layout reasons. XML comments are never visible in a publication.

```
<!-- This is an XML comment. -->
```

For information about formatting XML comments, see [Chapter 10, Formatting XML](#).

9.3 Entities

Entities are used to expand text. There are several situations in which they can be used:

- To represent special characters that cannot easily be displayed, entered or memorized.
- To integrate external files using entities representing references to their file names.
- To repeat content easily.

When an entity is defined, it can be used in many places. Entities increase consistency, as they only need to be defined once and will automatically be expanded everywhere within the document.

9.3.1 Common types of entities

Official generic entities are maintained in the [Doc Kit repository \(https://github.com/openSUSE/doc-kit\)](https://github.com/openSUSE/doc-kit). They include SUSE product names and other terms that are valid for every repository. In repositories configured with Doc Kit, the file `generic-entities.ent` therefore must not be changed (any changes will be overwritten by the next Doc Kit run). If there is a need to declare a specific entity that applies to the current repository only, this can be done in the `product-entities.ent` or `entity-decl.ent` file in the respective repository.

A `generic-entities.ent` or `entity-decl.ent` file contains several categories of entities:

Products

All SUSE product names and other products and applications. This helps when sudden name changes are necessary and avoids misspellings.

Platforms

Use entities for all hardware architectures referenced. This helps when sudden name changes are necessary.

Books

Title entities for all SUSE books. This helps when sudden name changes are necessary.

General Entities

Network IP addresses, host names and user names.

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

Entity selection

Use the entity name `&productname;` to identify the product for which the documentation is built. Set the value of this entity once per release and have it expand to the name of the current product:

```
&productname; includes 389-ds.
```

If you need to reference a specific subproduct or a different product, use a more specific entity:

```
Tuning &sle; for SAP
```

Acronyms

In some cases (for example, limited space in table cells or in titles), it is acceptable to use an entity for a product name acronym. Find the approved entities for product name acronyms in the entity declaration files, such as `product-entities.ent` or `generic-entities.ent`. For a product name acronym, you can use the generic entity `&product-nameshort;`. If you need acronym entities for specific products, they usually have an appended `a` at the end, for example, `&slsa;` for the acronym “SLES.”

Trademarks

Follow the rules under [Section 6.21, “Trademarks”](#) and [Section 7.17, “Products”](#).

If an entity is placed at the beginning of a phrase or title, its resolved form is usually lowercase (unless specified in uppercase, for example, for product names). To have it capitalized, use the `<phrase role="style:sentencecase"/>` element. For example:

```
<phrase role="style:sentencecase">&ulp;</phrase>
```

The entity `&ulp;` expands into “User space live patching”.

Never add this tag to the content within `<command/>` and `<systemitem/>` elements. Capitalizing anything inside these elements makes the content ambiguous.

9.3.2 Using entity files

SUSE uses a set of custom entities. Find these entities in the `*.ent` files in each documentation repository. One entity file can include other entity files.

- Entity files are only used for original, English-language documents. Translated documents contain only the resolved form of entities, that is, plain-text directly in the document.
- If you need a new entity to be used generically across all repositories, open a pull request against `generic-entities.ent` in the Doc Kit repository. After the change is approved by the Doc Kit maintainers, the entity update for `generic-entities.ent` will be rolled out to all Doc Kit-based repositories with the next Doc Kit run. If you need a custom entity that only applies to a specific repository, define it in `product-entities.ent` or in `entities-decl.ent` in this specific repository.

Do not include custom entity definitions directly in the file header, unless the custom definitions are needed to override externally embedded entities.

- Use the UTF-8 encoding when editing and saving the entity declaration file or any of the SUSE XML files.

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

```
<!ENTITY % entities SYSTEM "generic-entities.ent">
%entities;
```

EXAMPLE 9.1: EXCERPT FROM `product-entities.ent`

```
<!ENTITY ❶ product-sp ❷ "1❸">
<!ENTITY product-ga "15">
<!ENTITY productnumber-regurl "&product-ga;.&product-sp; ❹">
<!ENTITY productnumber "<phrase xmlns='http://docbook.org/ns/docbook'
role='productnumber'>15</phrase">❺">
```

- ❶ Making an entity declaration.
- ❷ Defining the entity name.
- ❸ Setting the value which the processed entity should expand to.
- ❹ Using another entity within the entity value. This entity reference is only valid if the other entity is defined somewhere within the scope of the XML document that is built. However, it does not matter whether the entity is defined before or after the current entity definition.
- ❺ Using a DocBook/GeekoDoc element within the entity value. The attribute `xmlns` must be included to define the correct XML namespace.

9.4 XInclude elements

XInclude elements are used to create modular source files that are easier to work with and can be reused. When editing a book, create a new source file for every chapter. Later, create a new GeekoDoc file that can serve as the central point. In this file, use XInclude elements to reference all chapters in the correct order:

```
<xi:include xmlns:xi="http://www.w3.org/2001/XInclude" href="gfdl.xml"/>
```

XInclude elements allow adding common sections to multiple books or articles without having to maintain the text in multiple places. Common sections include licenses and information on typographical conventions. XIncludes also simplify co-editing documentation with others in a version control system as they reduce the chance of merge conflicts.

By default, files referenced via XIncludes must be well-formed XML files that are also valid GeekoDoc fragments. This means that they must have a single top-level element and must not contain elements that are not allowed in GeekoDoc. Files that are supposed to be referenced multiple times from within the same set, book or article must not contain any `xml:id` attributes.

XIncludes also allow embedding plaintext files, for example, as the content in `<screen/>` elements. To embed a plaintext file, add the attribute `parse="text"` to the `<xi:include/>` element.

10 Formatting XML

This section provides information on formatting XML sources.

- **Line ends:** Lines should end with a Unix line end character (also called line feed, LF, newline, or `\n`).
- **Line length:** Lines should be at most 100 characters long, unless one of the following exceptions applies:
 - Some computer output, computer input or URIs may run longer and cannot be broken.
 - Some elements become much harder to read if broken. For example, that can be the case for long `<menuchoice/>` elements.
 - Aim to minimize the size of diffs in version control, to make reading diffs more efficient and version control storage more efficient. For example, if a typo fix introduces a line with 82 characters, consider keeping the line at that length. Also, avoid reflowing entire paragraphs of text, as that will also lead to hard-to-read diffs.
- **Indentation:** Indent using two space characters per indentation level. Make sure your editor does not replace spaces with tabs. Documents should not contain any tab characters.
- **Trailing whitespace:** Avoid introducing trailing whitespace characters such as spaces, protected spaces or tabs. Many editors have an option to view such characters. `git diff` will show newly introduced trailing whitespace characters in red.
- **Formatting of block elements:** Block elements are all DocBook elements that create a rectangular visual block in the layout, such as `<para/>`, `<table/>` or `<figure/>`. Format block elements with new lines before and after each tag and make sure they follow the indentation of the document:

```
<block>
  Content of block, indented by a space.
</block>
```

For information about the usage of `<screen/>`, see [Section 7.4.6, "Screens"](#).

- **Formatting of inline elements:** Block elements are all DocBook elements that can be re-flowed freely within a block element, such as `<emphasis/>`, `<keycap/>`, or `<guimenu/>`. Format inline elements along with other inline content without adding newlines or extra indentation:

```
<block>
  Content of block <inline>content of inline</inline>.
</block>
```

- **Formatting of title elements:** The title elements `<title/>` and `<term/>` are block elements. However, they provoke a style sheet quirk and should be treated differently. This avoids superfluous whitespace when used in the context of a cross-reference. Format title elements with new lines before the opening tag and after the closing tag and make sure they follow the indentation of the document:

```
<title>Content of Title</block>
```

- **Formatting of computer output/computer input blocks:** The Computer Output/Computer Input Block Element `<screen/>` should be treated like a block element but multi-line `<screen/>` elements should not be indented to aid source reading flow and avoid the trap of adding extraneous leading space to their content. (Single-line `<screen/>` can be indented).
- **XML comments.** XML comments should follow the indentation of the document. Where feasible, put XML comments on new lines to make reading diffs and later removal of the comment easier:

```
<block>
  Block content.
  <!-- An XML comment -->
</block>
```


XML comments must not contain the characters `--`. To preserve such character combinations within comments, replace them with `-/-`.


For information about the usage of XML comments, see [Section 9.2, "XML comments"](#).

- **Reflowing entire files:** Before reflowing entire files to a different XML formatting style, weigh the cost of keeping the document in its current state against the cost of reflowing. Reflowing often makes it easier to work with the document. However, if the document has a rich editing history already, reflowing makes it harder to properly use tools like **git blame**.


To easily convert documents to this style after importing them, use the commands **daps xml-format** (for an entire document) and **daps-xmlformat** (for a single file). Note that while these tools are very good otherwise, they do not reflow XML comments properly.

11 Working with AsciiDoc

To create documentation in the AsciiDoc format, adhere to the comprehensive guide at <https://asciidoctor.org/docs/asciidoc-recommended-practices> .

We also recommend the guidance on AsciiDoc provided in the [SUSE Technical Reference Documentation Contributors Guide](https://documentation.suse.com/trd/contributors/single-html/suse-trd_contrib-guide/#id-asciidoc) (https://documentation.suse.com/trd/contributors/single-html/suse-trd_contrib-guide/#id-asciidoc) .

The following things are important when working with AsciiDoc:

- Only use formatting that is supported by the AsciiDoctor tool. Ignore features that are only documented for the outdated **asciidoc** (Python) tool. In particular, ignore the documentation at <https://powerman.name>.
- Most recommendations from <https://github.com/SUSE/doc-susemanager/wiki/markup-syntax>  are applicable generally. Some recommendations, however, are specific to SUSE Manager documentation, in particular:
 - The section *Images*—images need to be added the same way they are added in other DAPS-based documentation, under the [images/src/FORMAT](#) directory of the repo.
 - The section *Working with Drafts*—there is currently no equivalent standard functionality.

12 Working with Smart Docs

Smart Docs represent a modular approach to documentation whose goal is producing sets of adaptable, solution-based, easy-to-find and reusable documents.

Rather than covering a complete technology or a set of problems, each Smart Docs article serves one distinct purpose. Each article stands for itself, including requirements, context, instructions, troubleshooting and FAQs.

Smart Docs consist of topics—smaller information units that can be reused in different contexts. Four different topic types are provided:

- Task topics instruct the user how to perform certain tasks.
- Concept topics explain underlying concepts to the users.
- Reference topics inform the user about basic facts (settings, options, parameters, etc.).
- Glue topics help navigate complex topics and provide links to related information (combine texts or structures that do not fit into any other category). Typical glue topics include “For more information” and “What's next” sections. Use glue topics to provide an additional layer of navigation for your article.

Articles are built by bundling these topics into assembly files—organizational units to structure the topics to create a coherent and meaningful document.

To create a Smart Docs project, use the templates for the assembly and its topical components that are provided in the [doc-modular repository \(https://github.com/SUSE/doc-modular/tree/main/templates\)](https://github.com/SUSE/doc-modular/tree/main/templates).

12.1 Using ITS tags with assemblies

Assembly files use ITS tags and attributes which flag whether the content of the tag should be translated or not. In the following example, the ITS attribute indicates that the content of the tag should not be translated:

```
<meta name="maintainer" content="Smart Doc author" its:translate="no"/>
```

If the `its:translate` attribute is set to `yes`, translation is needed.

Element name	Description	Translation
<code><meta name="bugtracker" its:translate="no"></code>	SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.	No
<code><phrase role="url">https://bugzilla.suse.com/enter_bug.cgi</phrase></code>	SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.	No
<code><phrase role="component">Non-product-specific documentation</phrase></code>	SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.	No
<code><phrase role="product">SmartDocs</phrase></code>	SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.	No
<code><phrase role="assignee">assigned@suse.com</phrase></code>	SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.	No
<code><meta name="translation" its:translate="no"/></code>	SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.	No
<code><phrase role="trans">yes</phrase></code>	SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.	No
<code><phrase role="language">de,cs-cz</phrase></code>	SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.	No
<code><meta name="architecture" content="x86;power"></code>	SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.	No

Element name	Description	Translation
<code>its:translate="no"/></code>		
<code><meta name="productname" its:translate="no"/></code>	The formal name of a product	No
<code><productname version="15-SP5">SLES</productname></code>	A number assigned to a product	No
<code><meta name="title" its:translate="yes">short title for SEO and social media, max. 55 chars</meta></code>	SEO-specific info. Adhere to the length limitations.	Yes Stick to the length limitations, min. 29 chars and max. 55 chars
<code><meta name="description" its:translate="yes">short description, max. 150 chars</meta></code>	SEO-specific info. Adhere to the length limitations.	Yes Stick to the length limitations, max. 150 chars
<code><meta name="social-descr" its:translate="yes">ultrashort description for social media, max. 55 chars</meta></code>	SEO-specific info. Adhere to the length limitations.	Yes Stick to the length limitations, max. 55 chars

Element name	Description	Translation
Doc Manager tags: <pre> <dm:docmanager xmlns:dm="urn:x- suse:ns:docmanager"> <dm:bugtracker> <dm:url>https:// bugzilla.suse.com/ enter_bug.cgi</ dm:url> <dm:component>Smart Docs</ dm:component> <dm:product>Documentation</ dm:product> <dm:assignee>maintainer@suse.com</ dm:assignee> </dm:bugtracker> <dm:editurl>https:// github.com/SUSE/ doc-modular/tree/ </dm:editurl> <dm:translation>no</ dm:translation> </dm:docmanager> </pre>	<p><u><info></u>: Wrapper to contain bibliographic information about the content and other meta info.</p> <p><u><dm:PLACEHOLDER></u>: SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.</p>	No
Resources: <pre> <resource xml:id="_glue- example" href=" ../ glues/glue.xml"> <description>Glue example</ description> </resource> </pre>	<p>SUSE-specific info needed by the doc-manager tool and DAPS to process and build SUSE user documentation.</p>	<p><u><resource></u> No</p> <p><u><description></u> Yes</p>

Element name	Description	Translation
Merge: <pre> <merge> <title>Your title, limit to 55 characters, if possible</title> <subtitle>Subtitle if necessary</ subtitle> <revhistory xml:id="rh-USE- ROOTID"> <revision> <date>2024-11-11</ date> <revdescription> <para> Describe the purpose of this revision </para> </ revdescription> </revision> <revision> <date>2024-10-10</ date> <revdescription> <para> Describe the purpose of this revision </para> </ revdescription> </revision> </revhistory> </pre>		<p><u><title></u> and <u><subti-</u> <u>tle></u> Yes, with length limi- tations (see above) <u><revhisto-</u> <u>ry></u> No <u><module></u> No <u><title></u> Yes</p>

Element name	Description	Translation
<pre><module resourceref="_concept- example" renderas="section"/ > </merge> <title>You are a very special concept now!</ title></pre>		

12.2 Revision history

A *revision history* lists all the high-level changes about the document itself. It gives the reader an overview of important changes made over time.

Update revision history regularly, mentioning most important changes to your document when you amend or rework it. The data you enter as revision history is used as metadata. The “Revision history” text is available as a link before the abstract and opens up in its individual page. Keep in mind the following rules:

- List revision entries in descending order by date. The latest entry must always come first.
- Describe only significant changes that you performed.
- If you have several changes of the same type (addition, change, removal), you may group them under a dedicated `<itemizedlist/>`.
- Do not add links or references with `xref` of any kind. If you want to mention a specific issue, use the abbreviations from https://en.opensuse.org/openSUSE:Packaging_Patch-es_guidelines#Current_set_of_abbreviations and wrap it inside the tag `uri`.

EXAMPLE 12.1: REVISION HISTORY EXAMPLE (SOURCE)

```
<revhistory xml:id="rh-USE-R00TID">
  <revision><date>YYYY-MM-DD</date>
    <revdescription>
      <para>Updated CONTENT, extended CONTENT, removed CONTENT</para>
    </revdescription>
  </revision>
  <revision><date>YYYY-MM-DD</date>
```

```
<revdescription>
  <para>Updated for the initial release of <phrase role="productname">SUSE Linux
Enterprise Server</phrase>
  <phrase role="productnumber">15 SP6</phrase></para>
</revdescription>
</revision>
</revhistory>
```

EXAMPLE 12.2: REVISION HISTORY EXAMPLE (OUTPUT)

Revision History

2024-11-13

Updated chapter on “foo bar”, removed section on “foo bar 1”

2024-09-21

Updated for the initial release of SUSE Linux Enterprise Server 15 SP6

A Terminology and general vocabulary

The following two tables define technical terms and general vocabulary for use in SUSE documentation. See also [Chapter 6, Language](#).

If unable to find a term below, check SUSE's official terminology database, [TermWeb \(https://suse.termweb.eu/\)](https://suse.termweb.eu/) [↗](#). It contains company-specific terminology in English and all our supported languages. TermWeb can be accessed only by SUSE employees and external partners with a SUSE account (get one at <https://www.suse.com/account/create> [↗](#)).

A.1 Terminology

The following table defines the correct spellings and denominations for technical terms in SUSE documentation. Always use the entry listed under “Accepted” in the table below. All terms are reproduced in sentence-style capitalization.

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
32-bit	32 Bit, 32 bit, 32-Bit, 32Bit, 32bit	adjective
3D	3 D, 3 d, 3.D., 3.d., 3-D, 3-d, 3d, Three-D	adjective
64-bit	64 Bit, 64 bit, 64-Bit, 64Bit, 64bit	adjective
AArch64	ARM64, ARMv8	noun; processor architecture
(to) activate sth.	(to) block sth., (to) check sth., (to) choose sth., (to) highlight sth., (to) tick sth.	verb; when referring to check boxes
adapter	adaptor	noun
add-on	add on, AddOn, addOn, addon	noun
address book	addressbook	noun
advice	advise [misspelling]	noun

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
(to) advise sth.	(to) advice sth. [misspelling]	verb
AMD64/Intel 64	x64, x86_64, x86-64, 64-bit AMD/Intel, AMD/Intel64	noun; processor architecture; see also x86
AOO	Aoo, aoo, OO, oo	noun; when referring to versions 3.4 and after; spelling according to project standard; acronym of <i>Apache OpenOffice</i> ; see also <i>OOo</i>
Apache OpenOffice	Apache Open Office, Apache Openoffice, OpenOffice	noun; when referring to versions 3.4 and after; spelling according to project standard; acronym is <i>AOO</i> ; see also <i>OpenOffice.org</i>
architecture	arch	noun; hardware platform, especially concerning processor platform
appendixes	appendices	noun; plural of appendix
application		noun; a computer program designed for a specific task or use
audio CD	Audio CD, Audio-CD, CD-Audio, CD Audio, CD audio	noun
back-end	back end, backend	noun
(to) back sth. up	(to) backup sth.	verb
backup	back-up, back up	noun
bare metal	bare-metal, baremetal	noun; a computer without an operating system; <i>also</i> a phys-

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
		ical computer (in contrast to a virtualized system)
bare-metal	bare metal, baremetal	adjective
Bash	BASH, bash	noun; spelling as per the GNU Bash manual
Bluetooth	Blue tooth, blue tooth, Bluetooth, blue-tooth, bluetooth	noun
Bluetooth card	wireless card [card has wires attached to it]	noun; card that enables Bluetooth connections.
boot disk	boot disc [usually a misspelling], boot-disk, bootdisk	noun; disk for starting the system
boot loader	boot-loader, bootloader	noun
(to) boot using PXE <i>or</i> (to) boot via PXE	(to) PXE boot	verb
Btrfs	B.T.R.F.S., Better FS, BetterFS, Butter FS, ButterFS, btrfs	noun; not an acronym
cursor	pointer [used for pointing device input]	noun; on-screen item indicating the position of keyboard input focus; see also <i>pointer</i>
CA	C.A., Ca	noun; acronym for <i>certificate authority</i>
CD	C.D., Cd	noun; acronym for <i>compact disc</i>
CD-ROM	CD ROM, CD-Rom, CD Rom	noun; acronym for <i>compact disc read-only memory</i>

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
CUPS	C.U.P.S., Cups, cups	noun; spelling as per project standard; acronym for <i>Common Unix Printing System</i>
case-sensitive	case sensitive, casesensitive	adjective
case-insensitive	case insensitive, caseinsensitive	adjective
certificate authority	certification authority, certifying authority, certified authority	noun; acronym is <i>CA</i>
changelog	change log, change-log, ChangeLog	noun; log of changes to software
check box	check-box, checkbox, checking option, tick box	noun; avoid, only mention name of option
checklist	check list, check-list, ticklist	noun
check mark	check, check-mark, checkmark, tick, tick mark	noun
chipset	chip set, chip-set	noun
(to) click sth.	(to) click on sth., (to) click onto sth.	verb; using a mouse button, usually to manipulate user interface element; also see <i>press</i>
client/server	client server, client-server, client–server, client + server	noun/noun
(to) close sth.	(to) abort sth. [negative], (to) exit sth., (to) kill sth., (to) terminate sth.	verb; when referring to closing a window; always use <i>quit</i> when ending an application normally; always use <i>terminate</i> when ending an application forcefully

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
codestream	code stream	noun; stream of code
(to) coldplug sth. (into sth.)	(to) cold plug sth. (into sth.), (to) cold-plug sth. (into sth.), (to) coldadd sth., (to) coldswap sth.	verb; adding a component or device to a system while the system is off
coldplugging	cold plugging, cold-plugging, coldadding, coldswapping	noun
coldpluggable	cold pluggable, cold-pluggable, coldaddable, coldswappable	adjective
Common Unix Printing System	Common UNIX Printing System, common Unix printing system	noun; spelling as per project standard; acronym is <i>CUPS</i>
command		noun; a signal that initiates an operation defined by an instruction
command line	command-line, commandline	noun
command-line	command line, commandline	adjective
configuration	config	noun; unless when referring to file extension
(to) configure sth.	(to) config sth.	verb
(to) connect via SSH (to sth.)	(to) connect by SSH (to sth.), (to) connect over SSH (to sth.), (to) connect through SSH (to sth.), (to) connect with SSH (to sth.), (to) SSH (to sth.), (to) ssh (to sth.), (to) ssh in (to sth.), (to) ssh into sth.	verb

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
console		noun; a physical terminal, used to describe TTYs and PTYs and when talking about consoles (e.g. KVM's console); see also <i>terminal</i> , <i>shell</i>
control center	Control Center, Control center, Control-Center, Control-center, control-center, Controlcenter, controlcenter	noun; generic term, as in: “the YaST control center” or “the KDE control center”
crash dump	crashdump	noun
(to) create a hard link (to sth.)	(to) hard link (sth.), (to) hardlink (sth.)	verb; see also <i>hard link</i>
(to) create a symbolic link (to sth.)	(to) soft link (sth.), (to) softlink (sth.), (to) symbolic link (sth.), (to) symlink (sth.)	verb; see also <i>hard link</i>
(to) deactivate sth.	(to) deblock sth., (to) uncheck sth., (to) untick sth.	verb; when referring to check boxes
delta RPM	delta-RPM, deltarpm	noun; RPM package that only includes files that changed between a previous and the current version of the package
(to) deselect sth.	(to) de-select sth., (to) remove the selection from sth., (to) un-select sth., (to) unselect sth.	verb; when referring to list entries or text; for check boxes, use <i>deactivate</i>
DHCP	D.H.C.P., Dhcp, dhcp	noun
dial-up	dial up, dialup	only as an adjective

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
dialog	dialog box, dialog window, dialogue [British], mask [Germanism], screen	noun; a secondary window that gives users progress feedback, prompts users to perform a command, enter information or select an option
directory	dir, folder	noun
DNS	D.N.S., DNS name server, Dns, dns	noun; acronym for <i>dynamic name server</i>
(to) double-click sth.	(to) double click sth., (to) double-click on sth., (to) double-click onto sth., (to) doubleclick sth.	verb
drop-down list	combination box, combo box, combobox, dropdown, drop-down, drop-down menu, drop-down list box, popover, pull-down menu	noun; GUI element; the list that is opened when clicking on it, showing a list of menu items the user can choose from; if list entries start actions, use <i>menu</i> instead
DVD	D.V.D., Dvd	noun; acronym for <i>digital versatile disc</i>
dynamic name server	Dynamic Name Server, Dynamic name server	noun; acronym is <i>DNS</i>
e-book	E-book, E-book, Ebook, eBook, electronic book, ebook	noun
e-mail	E-mail, E-mail, Email, eMail, electronic mail, email	noun
EPUB	E-PUB, e-PUB, e-Pub, EPub, Epub, ePUB, ePub	noun; project logo uses the capitalization “ePub,” but the vendor standard is “EPUB”

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
end user	end-user	noun; avoid; where possible, replace with <i>user</i>
(to) enter sth. (into sth.)		verb; only when a value needs to be specified and Enter should be pressed afterward; where possible, replace with <i>specify</i> or <i>provide</i>
Ethernet	ethernet	noun
Ethernet card	wired card [sounds as if wires attached to the card are meant]	noun; card that connects to networks via Ethernet
Ext3	EXT3, EXT 3, Ext 3, Ext-3, ext 3, ext-3, ext3	noun; use this capitalization for all versions of the Ext file system standard; intentionally inconsistent with project standard to emphasize that this is a proper name
Ext4	EXT4, EXT 4, Ext 4, Ext-4, ext 4, ext-4, ext4	noun
file name	file-name, filename	noun
file server	file-server, fileserver	noun
file system	file-system, filesystem	noun
flavor	flavour [British]	noun
flash drive	flash disk, flash disc, USB disk, USB drive, USB stick	noun
form		noun; a structured window, box or screen that contains

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
		numerous fields or spaces to enter data
framebuffer	frame buffer, frame-buffer	noun
front-end	front end, frontend	noun
FTP	F.T.P., Ftp, ftp	noun
GIMP	G.I.M.P., Gimp, gimp	noun; spelling as per project standard; acronym for <i>GNU Image Manipulation Program</i> ; if “the” occurs directly before <i>GIMP</i> , capitalize it: “The”
GNOME	G.N.O.M.E., GNU Networked Object Model Environment, Gnome	noun; spelling as per project standard; not an acronym
GRUB	G.R.U.B., Grub, grub	noun; acronym for <i>GRand Unified Bootloader</i>
graphical user interface	Graphical User Interface	noun; acronym for <i>graphical user interface</i>
GUI	G.U.I., GUI interface, GUI user interface, Gui	noun; acronym for <i>graphical user interface</i>
hard disk	HDD, HD, hard disc [misspelling], hard disk drive, hard drive, hard-disk, hard-drive, harddisk, harddrive, hdd, hd	noun
hard link	hard-link, hardlink	only as a noun; as a verb, use <i>create a hardlink link</i> ; directory entry that contains an alternative name for an existing file, in contrast to that, sym-

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
		<i>bolic links</i> are themselves files which link to the name of another file
home page	home-page, homepage	noun
host name	host-name, hostname	noun
(to) hotplug sth. (into sth.)	(to) hot plug sth. (into sth.), (to) hot-plug sth. (into sth.), (to) hotadd sth., (to) hotswap sth.	verb; adding a component or device to a system while the system is running; use <i>remove at runtime</i> where the specific action of removing a component or device is concerned
hotplugging	hot plugging, hot-plugging, hotadding, hotswapping	noun
hotpluggable	hot pluggable, hot-pluggable, hotaddable, hotswappable	adjective
HTML page	HTML document, HTML Web page, HTML web page	noun; when referring to a local file; see also <i>Web page</i>
HTTP	H.T.T.P., Http, http	noun
HTTPS	H.T.T.P.S., Https, https	noun
hypervisor	hyper visor, hyper-visor, hypervisor	noun
indexes	indices	noun; plural of index
infrared	infra red, infra-red	noun or adjective.
init script	init-script, initscript, initialization script [incorrect, when referring to script run by <u>init</u>]	noun; a script run by <u>init</u> ; for systemd, use <i>unit</i> or <i>unit file</i>

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
initialization	init, initialisation [British]	noun
(to) initialize sth.	(to) init sth., (to) initialise sth. [British]	verb
installation medium	installation data medium	noun; often in plural, “installation media”; only for <i>physical</i> sources of installation data for products; when physicality of the installation source is unclear or unimportant, use the more versatile term <i>installation source</i>
installation source	installation data source	noun; source of installation data for products, can be a physical medium or online repository
Internet	internet	noun
intranet	Intranet	noun
I/O port	I.O. port, I-O port, IO port, Io port	noun
IA64	IA-64, ia64, ipf, Itanium	noun; processor architecture
IPsec	IPSEC, Ipsec	noun
IPv4	IP v4, IPV4, Ipv4	noun; acronym for <i>Internet protocol version four</i>
IPv6	IP v6, IPV6, Ipv6	noun; acronym for <i>Internet protocol version six</i>
journaling	journalling [British]	noun

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
KIWI	Kiwi, kiwi	noun; project spelling; not an acronym; software for creation of operating system images
K Desktop Environment	Kool Desktop Environment	noun; spelling according to project standard; acronym is <i>KDE</i>
KDE	KDE Desktop Environment, K.D.E., Kde, kde	noun; spelling according to project standard; acronym for <i>K Desktop Environment</i>
Kdump	KDUMP, kdump	noun; only for application
kdump	KDUMP, Kdump	noun; only for command
kernel space	kernel-space, kernelspace, kernelland	noun; memory area reserved for the kernel and device drivers; see also <i>user space</i>
key combination	hot key, key accelerator, keyboard accelerator, key combo, keyboard combo, keyboard combination, key shortcut	noun
Kprobes	kprobes	noun; only for application
kprobes	Kprobes	noun; only for command
(to) left-click sth.	(to) click the left mouse, (to) click the left mouse button, (to) left click sth., (to) left-click on sth., (to) left-click on-to sth., (to) leftclick sth.	verb
left click		noun

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
LibreOffice	Libre Office, Libreoffice, Li-bO, LO, libreoffice	noun; spelling according to project standard; do not create acronyms of <i>LibreOffice</i>
license	licence [British]	noun
(to) license sth.	(to) licence sth. [British]	verb
lifecycle	life cycle, life-cycle	noun; a series of development and support stages that a product passes through
Linux	LINUX, linux	noun; spelling according to project standard
list	list field	noun; GUI element showing a list of menu items the user can choose from
live CD	LiveCD, live-CD	noun; CD that allows booting an operating system without installing
live DVD	LiveDVD, live-DVD	noun; DVD that allows booting an operating system without installing
live image	live disk image, LiveImage, live-image	noun; disk image that can be copied to a medium and then allows booting an operating system without installing
local host	local-host, localhost	noun; when describing the concept of hosting locally
localhost	local host, local-host	noun; when referring to the default name of a local host
log file	log-file, logfile	noun

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
login	log in, log-in	noun
logout	log out, log-out	noun
(to) log in [see below for appropriate preposition]	(to) log-in, (to) login, (to) log on, (to) log-on, (to) logon, (to) sign in, (to) sign on	verb
(to) log in to sth.	(to) log in at sth., (to) log into sth.	verb; for logging in to a device, application, etc.
(to) log in on sth.	(to) log in at sth., (to) log in from sth.	verb; for logging in on the console/a host system
(to) log in (to sth.) via SSH	(to) log in (to sth.) by SSH, (to) log in (to sth.) over SSH, (to) log in (to sth.) through SSH, (to) log in (to sth.) with SSH, (to) SSH (to sth.), (to) ssh (to sth.), (to) ssh in (to sth.), (to) ssh into sth.,	verb
(to) log out [see below for appropriate preposition]	(to) log off, (to) log-out, (to) logout, (to) sign off, (to) sign out	verb
(to) log out of sth.	(to) log out at sth., (to) log out from sth.	verb
loopback device	loop back device, loop-back device	noun
lowercase	lower case, lower-case	noun
mail server	mail-server, mailserver	noun
Maildir	Mail dir, mail dir	noun; specific format for e-mail storage, not a directory for e-mails

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
mainboard	main board, main-board, mother board, mother-board, motherboard	noun
man page	manual page, Man page, Man-page, man page, man-page, manpage	two words
Mbox	mbox	noun; specific format for e-mail storage
menu	drop-down menu	noun; GUI element that is a list whose entries each start an action; see also <i>drop-down list</i>
metadata	meta data, meta-data, meta-datas [misspelling]	noun
(to) middle-click sth.	(to) click the middle mouse, (to) click the middle mouse button, (to) middle click sth., (to) middle-click on sth., (to) middle-click onto sth., (to) middleclick sth.	verb
middle click		noun
mount point	mount-point, mountpoint	noun
mouse button	mouse-button, mousebutton, mouse key, mouse-key, mousekey	noun
(to) multitask	(to) multi task, (to) multi-task	verb
multitasking	multi tasking, multi-tasking	noun
multiuser	multi user, multi-user	noun

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
multi-version	multi version, multiversion	adjective
name server	name-server, nameserver	noun
namespace	name space, name-space	noun
need to	have to	verb; see also <i>must</i>
NFS	N.F.S., NFS file system, Nfs	noun; often: “NFS client,” “NFS server”
NIS	N.I.S., NIS information service, Nis	noun; often: “NIS client,” “NIS server”
OOo	Oo.o, Ooo, OOoo, OO, oo	noun; only when referring to versions prior to 3.4; spelling according to former project standard; acronym of <i>OpenOffice.org</i> ; see also <i>AOO</i>
(to) open sth.	(to) open up sth.	verb
OpenOffice.org	Open Office Org, OpenOffice, Openoffice.org, openoffice, openoffice.org	noun; only when referring to versions prior to 3.4; spelling according to former project standard; acronym is <i>OOo</i> ; see also <i>Apache OpenOffice</i>
openSUSE	Open SUSE, Open-SUSE, open SUSE, open-SUSE	noun; never capitalize first letter
open source	Open Source, Open-Source, open-source, opensource	only as a noun
operating system	operation system, operating-system	noun

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
paravirtualized	para-virtualised, paravirtualised [British], para-virtualized	adjective
patch level	patch-level, patchlevel	noun
path name	path-name, pathname	noun; avoid, check if <i>path</i> can be used instead
(to) plug sth. in	(to) plug-in sth., (to) plugin sth.	verb
plug-in	plug in, plugin	noun adjective
pointer	cursor [used for keyboard input], mouse cursor	noun; on-screen item echoing the movement of a pointing device, such as a mouse; <i>mouse pointer</i> is also acceptable; see also <i>cursor</i>
pop-up	pop up, popup	noun
on port	at port	noun with preposition
PostScript	POSTSCRIPT, Postscript, postscript	noun; spelling as per vendor standard
POWER	ppc64le, POWER8, Power	noun; processor architecture
(to) preconfigure sth.	(to) pre-configure sth.	verb
preconfigured	pre-configured	adjective
(to) print sth.	(to) print out sth.	verb
print queue	printer queue, printing queue	noun
print spooler	printer spooler, printing spooler	noun

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
(to) press sth.	(to) depress sth. [negative], (to) hit sth. [colloquial], (to) punch sth. [colloquial], (to) strike sth. [colloquial]	verb; when referring to keyboard keys or device buttons, but not mouse buttons; also see <i>click</i>
program		noun; a set of coded instructions that enables a machine, especially a computer, to perform a desired sequence of operations
proxy		only as a noun
PXE	P.X.E., Pixie, pixie, PXE Environment, Pxe, pxe	noun; acronym for “Preboot Execution Environment”
PXE boot	PXE Boot	only as a noun; as a verb, use “(to) boot using PXE” or “(to) boot via PXE” instead
(to) quit sth.	(to) abort sth., (to) exit sth., (to) kill sth., (to) terminate sth.	noun; quitting an application; always use “close” when referring to windows; always use “terminate” when ending an application forcefully
RAM	R.A.M., RAM memory, Ram, ram	noun; acronym for <i>random access memory</i>
RAM disk	RAM disc [misspelling], RAM drive, RAM-disk, RAM-drive, RAMdisk, RAM-drive, Ramdisk, Ramdrive	noun; either treating RAM as a hard disk or a type of solid-state storage
README	Read-me, Readme, read-me, readme	noun; use this capitalization for all general references

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
read-only	R.O., RO, read only, readonly, ro	adjective
real time	real-time	noun; as in “watch in real time”
real-time	real time	adjective, compound noun; as in “real-time processing”
(to) reconfigure sth.	(to) re-configure sth.	verb
(to) re-create sth.	(to) recreate [different meaning]	verb
(to) register [see below for appropriate preposition]	(to) sign up, (to) sign-up, (to) signup	verb; register as a user
(to) register at sth.		verb; register at a system
(to) register for sth.		verb; register for a service
(to) register to sth.		verb; use when writing about registering to a server: <i>register to the RMT server</i>
(to) register with sth.		verb; use when writing about the tool used for registration: <i>register the domain with libvirt</i>
(to) remove sth. at runtime (from sth.)	(to) hotremove sth.	verb; removing a component or device to a system while it is running; where sensible, use the more generic term <i>hot-plug</i>
(to) right-click sth.	(to) click the right mouse, (to) click the right mouse button, (to) right click sth., (to) right-	verb

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
	click on sth., (to) right-click onto sth., (to) rightclick sth.	
right click		noun
RPM	R.P.M., Rpm, rpm [different meaning]	noun; acronym for <i>RPM Package Manager</i>
runlevel	run level, run-level	noun
runtime	run time, run-time	noun
Samba	SAMBA, samba	noun; project spelling; open-source implementation of the SMB file and print service protocol
(to) save sth.	(to) store sth., (to) write sth. out	verb; when saving or overwriting a file from a GUI program or via a parameter of a command-line program; see also <i>write</i>
(to) save sth. as sth.		verb; when saving a file with a specific name
(to) save sth. in sth.		verb; when saving a file either on a specific device or file system
(to) save sth. on sth.		verb; when saving a file either on a specific device or file system
(to) save sth. to sth.		verb; when saving a file to a specific folder
saved in sth.		verb; when retrieving a file from a specific place

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
SCSI	S.C.S.I., Scsi, scsi	noun
screen		noun; the surface on which the image appears in an electronic display (as in a computer terminal); also the information displayed on a computer screen at one time
screenshot	screen shot, screen-shot	noun
screen saver	screen-saver, screensaver	noun
script		noun; a prewritten list of commands, and perhaps other control information, to be executed (interpreted) by a shell or other command interpreter
scrollbar	scroll-bar, scroll bar, scroll-box, scroller, slidebar	noun; GUI element that is used to change which portion of a screen area is visible
(to) select sth.	(to) block sth., (to) choose sth., (to) highlight sth.	verb; when referring to list entries or text; for check boxes, use <i>activate</i>
selected	blocked, chosen, highlighted	adjective; selection state of list entries or text; opposite of <i>deselected</i>
(to) set sth. up	(to) set-up sth., (to) setup sth.	verb
setup	set up, set-up	adjective noun
shell		noun; a command-line interpreter used to describe programs that expose the in-

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
		put/output to the user (bash, sh, zsh) and to refer to the command prompt; see also <i>console, terminal</i>
(to) shut sth. down	(to) shut-down sth., (to) shut-down sth.	verb
shutdown	shut down, shut-down	adjective noun
SLE	S.L.E., SLE Enterprise, SLE Linux, Sle, sle	noun; avoid; acronym for <i>SUSE Linux Enterprise</i>
SLED	S.L.E.D., SLE Desktop, SLE Enterprise Desktop, SLE Linux Desktop, Sled, sled	noun; avoid; acronym for <i>SUSE Linux Enterprise Desktop</i>
SLES	S.L.E.S., SLE Server, SLE Enterprise Server, SLE Linux Server, Sles, sles	noun; avoid; acronym for <i>SUSE Linux Enterprise Server</i>
SLES for SAP	SLES for SAP Applications, SLE for SAP	noun; acronym for <i>SUSE Linux Enterprise Server for SAP Applications</i>
slider	slide bar, sidebar	noun; GUI element that is used manipulate values that have an upper and a lower bound
solid-state drive	SD [misleading], solid state disc [misspelling], solid-state disk drive, solid-state disk, solid state drive, solidstate drive, sd	noun; acronym is <i>SSD</i> ; a type of mass storage that does not depend on mechanical parts
spec file	Spec file, Spec-file, Specfile, spec-file, specfile	noun

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
SSD	S.S.D., SD [misleading], SS-D, sd, ss-d	noun; acronym of <i>solid-state drive</i> ; a type of mass storage that does not depend on mechanical parts
stand-alone	stand alone, standalone	adjective
(to) start sth. up	(to) start-up sth., (to) startup sth.	verb
start-up	start up, startup	noun
statusbar	status bar, status-bar	noun
SSH	S.S.H., SSH Shell, SSH shell, Ssh, ssh	noun
SUSE	S.U.S.E., Software- und System-Entwicklung, SuSE, SuSe, Suse, suse	noun; not an acronym
SUSE Enterprise Storage	SUSE Storage, SUSE Linux Enterprise Storage	noun; acronym is <i>SES</i>
SUSE Linux Enterprise	SUSE Linux Enterprise [British], SUSE Linux enterprise, SUSE linux enterprise	noun; acronym is <i>SLE</i>
SUSE Linux Enterprise Desktop	SUSE Desktop, SUSE Linux Enterprise desktop	noun; acronym is <i>SLED</i>
SUSE Linux Enterprise Server	SUSE Server, SUSE Linux Enterprise server	noun; acronym is <i>SLES</i>
SUSE Linux Enterprise Server for SAP Applications	SUSE Linux Enterprise for SAP, SUSE Linux Enterprise Server for SAP, SUSE Server for SAP	noun; acronym is <i>SLES for SAP Applications</i>

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
SUSE Manager	SUSE Linux Manager	noun
SUSE OpenStack Cloud	SUSE Cloud, SUSE Linux Cloud	noun
SUSE Studio	SUSE Linux Studio	noun
submenu	sub menu, sub-menu	noun; <i>menu</i> that is nested inside another menu
subsystem	sub system, sub-system	noun
systemd	System D, Systemd, systemD, system d, System 500	noun; project spelling; initialization system for Linux
System V init	SysVinit, SysV init, system 5 init, system d	noun; spoken: “System five init”; initialization system for Unix operating systems
system-wide	systemwide, system wide	adjective
symbolic link	soft link, softlink, symlink [jargon]	only as a noun; as a verb, use <i>create a symbolic link</i> ; a file with a reference to another file or a directory, in contrast to that, a <i>hard link</i> is a directory entry that contains an alternative name for an existing file
synchronization	sync, synch, synchronisation [British]	noun; two-way or many-way copying process to ensure data is consistent across two or more locations
(to) synchronize sth. (with sth.)	(to) sync sth., (to) synch sth., (to) synchronise sth. [British],	noun; copy data in two or more ways to ensure it is con-

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
	(to) synchronize sth. (and sth.)	sistent across two or more locations
TAR archive	TAR ball [Unix jargon], tar ball, tar-ball, tarball	noun
taskbar	task bar, task-bar	noun
technology preview	technical preview, technology-preview	noun; product features that are shipped without support and marked as such
text box	entry area, entry box, entry field, input area, input box, input field, text area, text field	noun; GUI element that text can be typed into with one or more lines
terminal		noun; text input/output environment where users interact with Linux and Linux applications; the default term to describe a text-only user interface
(to) terminate sth.	(to) abort sth., (to) close sth., (to) exit sth., (to) kill sth., (to) quit sth.	noun; ending an application forcefully; always use <i>close</i> when referring to windows; always use <i>quit</i> when ending an application normally
TFTP	T.F.T.P., Tftp, tftp	noun
timeout	time-out	noun
time stamp	time-stamp, timestamp	noun
titlebar	title bar, title-bar	noun
tool		noun; a utility or feature used to develop software or hard-

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
		ware, or to perform particular tasks
toolbar	tool bar, tool-bar	noun
toolchain	tool chain, tool-chain	noun; set of tools (such as build tools) that is used in succession
tooltip	tool tip, tool-tip	noun
UEFI	Uefi, u-EFI, uEFI	noun; acronym of <i>Unified Extensible Firmware Interface</i>
Unified Extensible Firmware Interface	unified extensible firmware interface	noun; acronym is <i>UEFI</i> ; software interface between firmware and operating system; replaces the BIOS interface
unit	unit file	noun; <i>concept</i> of systemd; generic term for services, timers, etc. (https://www.freedesktop.org/software/systemd/man/systemd.unit.html) ; use when starting, stopping, enabling or disabling a unit
unit file	unit	noun; <i>configuration file</i> of a systemd unit; has a suffix (<i>.service</i> , <i>.timer</i> , etc.); only use when referring to the actual file (e.g. when editing it) and not the unit

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
Unix	UNIX [brand name registered by Open Group], unix	noun; use this capitalization for all general references that are not related to brand names
(to) uninstall sth.	(to) deinstall sth., (to) un-install sth.	verb
unselected	deselected, un-selected	adjective; selection state of list entries or text; opposite of <i>selected</i>
uppercase	upper case, upper-case	noun
usage		noun; the way in which something is used, or the amount of it that is used; see also <i>utilization</i>
use case	use-case, usecase	noun
(to) use sth.	(to) utilise sth. [British], (to) utilize sth.	verb
user name	user-name, username	noun
user space	user-space, userspace, user-land	noun; memory area used by applications; see also <i>kernel space</i>
utilization	utilisation [British]	noun; an act or instance of making practical or profitable use of something, especially in <i>CPU utilization</i> , <i>memory utilization</i>
video DVD	Video DVD, Video-DVD, DVD video	noun

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
view		noun; a reusable set of user interface widgets that serve as an interface for user interaction
virtualization	Virtualization, virtualisation [British]	noun; referring to software (usually an operating system) running on a virtual computer created by software running on a physical computer <i>or</i> virtual computer created with software running on a physical computer
(to) virtualize sth.	virtualise [British]	verb; running software (usually an operating system) on a virtual computer created by software running on a physical computer <i>or</i> creating a virtual computer with software running on a physical computer
VLAN	V.L.A.N., Vlan, vlan	noun; acronym for <i>Virtualized Local Area Network</i>
Web	WEB, World Wide Web, WWW, web, www	noun; you may use <i>World Wide Web</i> or <i>WWW</i> in historical contexts
Web cam	Webcam, Web camera, web-cam	noun; camera that can be connected to a computer, mainly for video chats

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
Web page	HTML Web page, Web-page, Webpage	noun; when referring to page on the Internet; see also <i>HTML page</i>
Web server	Web-server, Webserver	noun
Web site	Web-site, Website, web site, web-site, website	noun
Webmaster	Web master, Web-master	noun
whitespace	white-space, white space	noun
Wi-Fi	Wi fi, Wi-fi, Wifi, wireless fidelity, WLAN	noun; use the <i>Wi-Fi</i> brand name whenever referring to IEEE 802.11-based networks or access points; use <i>WLAN</i> when referring to non-IEEE 802.11-based wireless LANs
Wi-Fi card	wireless card [card has wires attached to it]	noun; card that connects to Wi-Fi networks
Wi-Fi/Bluetooth card	wireless card [card has wires attached to it]	noun; card that combines a Wi-Fi and a Bluetooth card
wild card	joker [Germanism], wild-card, wildcard	noun
WLAN	Wlan	noun; avoid; use only when referring to wireless LANs that are not IEEE 802.11-based; use <i>Wi-Fi</i> in all other cases
(to) write sth.	(to) pipe sth. [Unix jargon], (to) write sth. out	verb; when saving the command-line output of a pro-


Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
		gram as a file using <code>></code> or <code>>></code> ; see also <i>save</i>
x86	32-bit AMD/Intel, i686, i386	noun; processor architecture; see also <i>AMD64/Intel 64</i>
X Window System	X Window, X Windows, X window, X window system, X windows, XWS	noun
Xen	XEN, xen	noun
Xend	xend	noun
YaST	YAST, YAST2, Yast, YaST2, yast, yast2	noun; spelling according to project standard; acronym for <i>Yet another Setup Tool</i>
IBM Z	z Systems, System z, zSeries, z System, zsystems, S390x	noun; processor architecture; see also <i>AMD64/Intel 64</i>
Zypper	zypper	noun; only for application
zypper	Zypper	noun; only for command

A.2 General vocabulary

The following table defines the correct spellings and denominations for general vocabulary in SUSE documentation. Always use the entry listed under “Accepted” in the table below. All entries are reproduced in sentence-style capitalization.



Note: Review the word list of the Inclusive Naming Initiative

In addition to the words documented here, make sure to also review the Word lists of the Inclusive Naming Initiative's [Evaluation Framework \(https://inclusivenaming.org/language/evaluation-framework/\)](https://inclusivenaming.org/language/evaluation-framework/) .

For more information about word choices, see [Section 6.2, “Biases and inclusiveness”](#).

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
after	once	adverb; only use <i>once</i> in the meaning of “one time only”
afterward	afterwards [BrE]	adverb
although	while	conjunction; only use <i>while</i> in the meaning of “during the time that”
and	while	conjunction; only use <i>while</i> in the meaning of “during the time that”
backward	backwards [BrE]	adverb
	basically [filler]	adverb
because of	since, due to, owing to	preposition; only use <i>since</i> in temporal phrases
business case	business-case, businesscase	noun
but	while	conjunction; only use <i>while</i> in the meaning of “during the time that”
cannot	can't [contraction], can not	verb
can	may	verb; use <i>can</i> to express an ability, only use <i>may</i> to express permissions sought/given
could	may	verb; use <i>could</i> to express a possibility, only use <i>may</i> to express permissions sought/given

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
data is	data are	noun with verb; use all other verbs in the singular
	easy [filler], easily	adjective, adverb; avoid
etc.		abbreviation; avoid; do not use together with “for example” and “such as”; always precede with a comma
for example	for instance, for instances [misspelling]	adverb
forward	forwards [BrE]	adverb
if		pronoun; use if an event depends on a condition; also see <i>when</i> and <i>whether</i>
inward	inwards [BrE]	adverb
	just [filler]	adjective, adverb; avoid
might	may	verb; use <i>might</i> to express a possibility, only use <i>may</i> to express permissions sought/given
must	have to	verb; see also <i>need to</i>
need to	have to	verb; see also <i>must</i>
	obvious [insulting], obviously	adjective, adverb
outward	outwards [BrE]	adverb
	please	adverb; avoid
	self-evident [insulting], self-evidently	adjective, adverb

Accepted	Rejected [Reason]	Part of Speech; Usage Guideline/Definition
sideward	sidewards [BrE]	adverb
	simple [filler], simply	adjective, adverb; avoid
(to) simplify sth.	(to) ease sth., (to) facilitate sth.	verb; avoid
(to) simplify sth.	(to) ease sth., (to) facilitate sth.	verb; avoid
	stuff [colloquial], stuffs	noun
toward	towards [BrE]	adverb
want sth.	(to) wish sth., (to) wish for sth., would like sth.	verb
when	once	adverb; use <i>once</i> only in the meaning “one time only”
when		pronoun; use if an event is inevitable; also see <i>if</i>
whether	whether or not	pronoun; use to present two alternatives which are not conditions, otherwise use <i>if</i> ; see also <i>if</i>
regarding	as regards, in regard to, with regard to, with regards to	preposition

B Contributors

For a list of people who contributed to this document, visit [the *Contributors* page of our Git repository](https://github.com/SUSE/doc-styleguide/graphs/contributors) (<https://github.com/SUSE/doc-styleguide/graphs/contributors>) .

C GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

C.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. Secondly, 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.

C.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 Transpar-

ent 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 lan-

guage. (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.

C.3 VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, 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.

C.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 com-

puter-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.

C.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 His-

tory 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.

C.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”.

C.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.

C.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.

C.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.

C.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.

C.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.

C.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.