

Comprehensive, hands-on training that solves real world problems

English Lessons for Tech Writers

Student Workbook

ENGLISH LESSONS FOR TECH WRITERS

Documentation 0.2 English for Techies

English Lessons for Tech Writers

Edition 1 0

Authors: Seth Kenlon , David O'Brien

Copyright © 2017 Red Hat, Inc.

The contents of this course and all its modules and related materials, including handouts to audience members, are Copyright © 2017 Red Hat, Inc.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Red Hat, Inc.

This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat, Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.

If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed please e-mail training@redhat.com or phone toll-free (USA) +1 (866) 626-2994 or +1 (919) 754-3700.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, Hibernate, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a registered trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

This class covers common grammar that all writers should be familiar with, some writing practices specific to Red Hat, and some specific to GLS. All are important for an effective technical writer.

Preface	vii
Document Conventions	vii
We Need Feedback!	xi
1. Optimizing Sentences for Efficiency	1
Implementing Minimalism	2
Exercise	4
2. Writing in Active Voice	5
Writing in the Active Voice	6
Exercise	8
3. Writing Nouns and Verbs that Agree with Each Other	9
4. Definite and Indefinite Articles	11
Exercise	12
5. Pronouns	13
How to Use Pronouns	14
Exercise	17
6. Prepositions	19
Identifying Prepositional Phrases	20
Choosing the Correct Preposition	21
Exercise	22
7. Conjunctions	23
Transforming Conjunctions into Simple Sentences	24
Choosing the Right Conjunctions	25
Exercise	26
8. Punctuating Sentences Correctly	27
Period, Full-stop, Dot	28
Comma	29
Colon	30
Semicolon	31
Parentheses, Rounded Brackets	32
Question Marks	33
Quotation Marks	34
Exercise	35
9. Using the Correct Possessive	37
Converting Nouns into Possessive Nouns	38
Converting Pronouns into Possessive Pronouns	39
Exercise	41
10. Lists and Tables	43
Stems	44
Punctuation	45
Tense	46
Key-Value Lists	47
Tables	48
11. Creating Reference Sections	49
Referencing Red Hat Docs within an Admonition	50
Generic Referencing within an Admonition	51
Combining Multiple Reference Types	52

A. Revision History	55
Index	57

Preface

Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

Press **Enter** to execute the command.

Press **Ctrl+Alt+F2** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **mono-spaced bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog-box text; labeled buttons; check-box and radio-button labels; menu titles and submenu titles. For example:

Choose **System > Preferences > Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications > Accessories > Character Map** from the main menu bar. Next, choose **Search > Find...** from

the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit > Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or *Proportional Bold Italic*

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above: username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo             echo   = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled “Important” will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

We Need Feedback!

If you find an error in this manual, please contact curriculum-core@redhat.com.

CHAPTER 1

OPTIMIZING SENTENCES FOR EFFICIENCY

"Minimalism", in the context of technical writing, is the art of writing more clearly by writing less.

That definition, rewritten the *wrong* way:

"Minimalism" is an artistic discipline that can appear in any medium, even technical writing. When writing technical documentation, you may want to use minimalism. You should write fewer words in order to reach your central point, but without the loss of useful or important information.

The first version, while terse, is better. It is simple, it states exactly the information the author wants to convey, it leaves no room for questions. This is typical of minimalism.

Advantages of minimalism are:

- Writing fewer words provide fewer opportunities for you to make writing mistakes.
- Readers are less likely to misinterpret the intent of a message when your writing is direct and concise.
- Maintenance and updates are easier when there is less to maintain and update.
- Translation of your writing to other languages is easier when the meaning of your writing is obvious.
- Tech editing is faster when there are fewer words to review.
- It is more efficient to write less to communicate more.

Implementing Minimalism

Humans tend to think verbosely, so minimalism does not usually occur in a first draft. That's expected, but during your self-edit, look for sentences that use idiomatic phrases, "filler" words, or that restate a point that has already been made, and rewrite them using fewer words.

Here are some examples of how to use minimalism by rewriting verbose sentences:

Convert Paragraphs to Minimalism

Bad (21 words):

A static IP address should be set on the server in order to log in to it remotely from the client.

Better (12 words):

Set a static IP address on the server to enable remote login.

Bad (47 words):

The standard XYZ solution for the granularity concern these concerns would be XYZ Bundle Repository (XBR), but XBR failed to get traction in the market. Another example of a popular software system that relies on XYZ but choose to not rely on XBR is the Emacs editor.

Better (20 words):

Emacs is an XYZ-compliant editor allowing developers to create and modify XYZ bundles without installing XYZ Bundle Repository.

Ideally, minimalism doesn't just shorten a sentence, but makes the point of the sentence clearer. It allows the reader to focus on what is important.

Exercise

Rewrite each paragraph, bearing minimalism in mind:

1. Click on the **OK** button to confirm the action and close the dialog box.
2. Look through each menu item to familiarize yourself with all the options available to you throughout the interface before attempting to utilize the API so that you have some idea of what functions users expect to have available to them.
3. Click on the **OX** button to confirm the action and close the dialog box.
4. In the right panel, you should see that the **XYZ** view will show that the most recent task has been completed and is waiting for you to continue. In order to verify this, click the **OK** in the lower left corner of the interface in your web browser. This will display a verification message confirming that the test passed and that it is safe for you to continue to the next phase of the exercise.
5. As you will see later in this chapter, both the **foo** and **bar** applications work with a pipeline concept, which is sequential. However, you may want to use **foo** when it's available, since it also has the ability to trigger an exchange to multiple destinations simultaneously, utilizing parallel processing where possible.

CHAPTER 2

WRITING IN ACTIVE VOICE

Writing in the *active voice* creates powerful and engaging content. The active voice suggests that the written word is "speaking" directly to the reader, enabling the reader to learn quickly, and to take action when an exercise requires it. For technical writing, the active voice is the clearest way to demonstrate action and result, or cause and effect.

Active voice

In the active voice, the subject of the sentence is also the person or thing performing an action.

I compiled the code.

Passive voice

The *passive voice* is the opposite of active voice. In the passive voice, the subject of the sentence is the person or thing receiving action:

The code was compiled by me.

While the passive voice is not grammatically incorrect, it is less efficient than the active voice and also tends to cause confusion about who is doing what.

Here is another sentence in the passive voice:

A key being pressed prompts the computer to continue.

Is this sentence an instruction telling you to press a key, or is it just a footnote about what happens when a key is pressed?

The active voice is both clearer and more concise:

Press any key to continue.

Writing in the Active Voice

To use the active voice, write as if you're composing a list of instructions for your reader. You are not writing a description of what the reader *would see* if the reader *were sitting* in front of your computer, seeing what you see as you write. The reader is seeing what is on the reader's own computer screen, and they are doing what you tell them to do.

Here are some examples of sentences written in the passive voice, and how to rewrite them in the active voice:

Convert Sentences to Active Voice

Passive voice:

The values of the XML configuration can be found by parsing the file with the Xerces library in C++.

Active voice:

Parse the XML configuration file using the *Xerces* [<https://xerces.apache.org/xerces-c/>] C++ library.

Passive voice:

Open **XYZ Viewer** by double-clicking the **XYZ Tools** icon from the workstation desktop.

Active voice:

Double-click the **XYZ Tools** icon on the workstation desktop to open **XYZ Viewer**.

Passive voice:

It was presented earlier in this book the syntax for using XML configuration files with a Python module imported into the **PyShadowman** code base.

Active voice:

The steps to import the required module into **PyShadowman** were explained in the *Importing the BeautifulSoup Module* [#] section.

Exercise

Rewrite each paragraph in the active voice.

1. The server got rebooted.
2. The QTextEdit field should be placed in the right corner of the QMainWindow.
3. To begin the code compilation process, GCC is used.
4. The previous three steps should be repeated to complete RHSM registration on each machine.
5. The virtual machine should be started before attempting to ping it.
6. The open source Mumble audio chat application should be used to provide technical support to users.

CHAPTER 3

WRITING NOUNS AND VERBS THAT AGREE WITH EACH OTHER

Nouns, pronouns, and verbs all have singular and plural forms. The subject of a sentence and its verb must *agree* on whether they are each singular or plural. When you write a singular noun coupled with a singular verb, or a plural noun with a plural verb, they are in agreement.

For example, this sentence uses a subject and verb that are in agreement, because the subject *hard drive* is singular, and so is its verb *is*:

Your hard drive is in the laptop.

This sentence demonstrates a subject and verb that do not agree, because the subject *hard drive* is singular but its *are* is plural:

Your hard drive are in the laptop.

Here are the rules for subject and verb agreement:

Subject and Verb Agreement

Singular subject

Use a singular verb for a singular subject.

The *httpd daemon* is active and enabled.

The verb must agree with the noun, regardless of a noun's adjectives or clauses. For instance, the noun *database* is singular and gets a singular verb, even though there are plural modifiers describing the subject:

A *database* of customers and sales people *is* stored on the server.

Either or

Use a singular verb when giving a choice between singular subjects.

Either *IPv4* or *IPv6* *is* supported by this project.

Compound singular subject

Use a plural verb for a sentence with more than one singular subjects.

The *httpd daemon* and the *rsync daemon* *are* active and enabled.

Plural subject

Use a plural verb for a plural subject.

The *emails* *are* encrypted.

Ansible projects and their playbooks are key tools in DevOps.

CHAPTER 4

DEFINITE AND INDEFINITE ARTICLES

The word *the* is a definite article. Use it to:

- refer to a specific instance of something. For example:

Click the **OK** button.

- refer to all instances of something. For example:

The **for** loop is a common programming construct.

The words *a* and *an* are indefinite articles. Use them to:

- refer to a classification of something, but not a specific instance of one. Two examples:

Press a key to continue.

Use a compiler to build this code.

- refer to something generically. For example:

An image from the video is displayed.

When the task is complete, a new window opens.

There are two indefinite articles. Which one you use depends on the object it precedes.

Use *a* when the object begins with a consonant, or sounds like it begins with a consonant when pronounced aloud:

a UNIX system

a window

a button

a TCP port

Use *an* when the object begins with a vowel, or sounds like it begins with a vowel when pronounced aloud:

an MPEG stream

an application

an error message

Exercise

Identify the incorrect articles in these sentences:

1. Log in to the workstation VM and open the window to `/home/student/code`.
2. Click an **Next** button to continue.
3. By default, there are three windows on the desktop. Read a title bar of each window to find the window called **Foo**. In this window, enter your password in a text field.

CHAPTER 5

PRONOUNS

Pronouns are pointers to a noun. These are some common singular pronouns:

- you
- it
- this
- that

These are some common plural pronouns:

- they
- these
- those

In technical writing, pronouns are simplified because English doesn't assign gender to inorganic things.

Note

If you are writing a use case and need to reference a fictional human, use generic terms such as a job title, like systems administrator or programmer, rather than an arbitrary name like "John" or "Jane". Use the pronoun "they" instead of "him" or "her".

How to Use Pronouns

For pronouns to work as designed, they require two things:

1. The reader must know what noun the pronoun references.
2. The tense of the pronoun and its noun and verb must match.

Here are some examples:

Singular noun

When referring back to a singular noun, use the pronoun *it*. For example:

The server is running Red Hat Enterprise Linux 7, and it has 8 TB of storage.

Plural pronouns

When referring back to a plural noun, use the pronoun *they*. For example:

Desktop computers often get used for office tasks, but they can also be configured as servers.

Possessive pronouns

To indicate that a pronoun possesses something, use the possessive form of the appropriate pronoun. For example:

Emacs is a text editor, but its real power is its LISP interpreter.

The same applies for plural forms:

Open source programmers are brave to subject their code to the scrutiny of anyone who wants to look at it.

You may have both singular and plural in a sentence:

Your colleagues may run Windows or MacOS on their computers, but you should run Linux on yours.

Pronouns in context

Sometimes a pronoun is used in a separate sentence than the one establishing the noun being referenced. This relies on the reader to understand from context which noun the pronoun most likely refers to.

Sometimes it is very obvious:

I like drinking coffee while I sit and type at my computer. Sometimes, I even dunk a doughnut into it.

In that phrase, the reader knows what "it" means from contextual clues. The writer means "Sometimes, I dunk a doughnut into my coffee", not "Sometimes, I dunk a doughnut into my computer."

Sometimes it is less obvious:

The **foo** command is being developed to replace the **bar** command. It is installed by default, and it has a bug in it that erases all of your data permanently.

In that paragraph, the reader cannot know definitively which command, **foo** or **bar**, erases all of their data and which command is safe to use. Sometimes, using a pronoun is not as effective as being explicit.

Try rewriting the sample paragraph, assuming that **foo** is the dangerous command and that **bar** is the safe command.

Verifying pronouns

To verify that a pronoun makes sense, reread a sentence or paragraph, replacing the pronoun you chose with the subject: For example:

Desktop computers often get used for office tasks, but they can also be configured as servers for a small network.

Desktop computers often get used for office tasks, but desktop computers can also be configured as servers for a small network.

If it does not make sense, or if it impossible to read back with its subject replaced because there is no indication of what the correct subject is, then the pronoun has not been used correctly.

The **foo** command is being developed to replace the **bar** command. It is installed by default, and it has a bug in it that erases all of your data permanently.

The **foo** command is being developed to replace the **bar** command. *Foo? Bar?* is installed by default, and *foo? bar?* has a bug in *foo? bar?* that erases all of your data permanently.

If it still says what you meant to say, then you have placed the pronoun in the correct place. Here is an example using a plural pronoun:

Pronouns are like variables. They hold whatever information the author places into them.

Pronouns are like variables. Pronouns and variables hold whatever information the author places into pronouns and variables.

Exercise

Correct these pronouns as needed:

1. The systems administrators imaged the computers so that its host names followed a logical naming scheme.
2. Return the laptop to the store and get refunded for them.
3. Okular is the best PDF viewer on the market. Admittedly, it's not the best format for digital releases compared to EPUB, but it's great for printing.
4. The systems administrators imaged each computer so that its host name followed a logical naming scheme.

CHAPTER 6

PREPOSITIONS

Prepositions answer the question "which one?", "what kind?", "how much?", or "how many?" about a noun. A preposition is a helper word for nouns, explaining which noun is being written about in a sentence.

Common prepositions include:

- on
- in
- from
- to
- at

For example, in these sentences, the prepositions are *emphasized*:

Each Linux user is mapped *to* an SELinux user *by* an SELinux policy.

After configuring the guest NUMA node topology, specify the huge page size and the guest NUMA nodeset *in* the <memoryBacking> element *in* the guest XML.

A prepositional phrase contains the object that relates to the noun being helped by the preposition. In the following examples, the prepositional phrases are in brackets ([and]):

Each Linux user is mapped [to an SELinux user] [by an SELinux policy].

After configuring the guest NUMA node topology, specify the huge page size and the guest NUMA nodeset [*in* the <memoryBacking> element] [*in* the guest XML].

Identifying Prepositional Phrases

To find a prepositional phrase, locate the preposition and then ask "what?". For example:

Each Linux user is mapped *to* what?

Each Linux user is mapped *to* an SELinux user *by* what?

Each Linux user is mapped *to* an SELinux user *by* an SELinux policy.

Choosing the Correct Preposition

In English, each preposition has a specific meaning. Using the wrong preposition confuses a reader, or even changes the meaning of a sentence. Choosing the right one to lead your prepositional phrase is important.

on

Literally, "on top of", as when a cat sits on a mat. In tech writing, "on" is used to express an action performed at a computer you have logged in to, and some GUI actions. For example:

Click the icon on the desktop.

Run the setup script on the workstation virtual machine.

in

Literally, "inside", as when a cat sits inside of a box, or when a number is found in a mathematical set. In tech writing, "in" is used to express an action performed to a non-binary file, or to show membership to a group or system or network or framework. For example:

Click the icon in the upper-right corner of the desktop.

Log in to the workstation virtual machine.

The user managing the foo service must be in the wheel group.

to

Literally, "toward", as when a cat walks to its food. The word "to" implies that you are moving in the direction of something not yet reached. For example:

Drag the file to the trash icon.

Upload the file to the server.

from

Literally, "away", as when a cat comes indoors from the garden. The word "from" implies that you have been in one place, and are now leaving that place behind. For example:

If you do not want to delete the file, move the file icon from the trash.

Download the file from the server so you can edit it.

Exercise

Identify the prepositions and prepositional phrases, if they exist, in each sentence:

1. The configuration file is installed in the **/etc** directory.
2. Set the required parameters in the fields under the **Performance options** panel.
3. The **foo** command instructs the kernel to pipe the core dump to the **abrt-hook-ccpp** program.
4. In Red Hat Enterprise Linux 7, the **systemctl** utility replaces a number of power management commands used in previous versions of the Red Hat Enterprise Linux system.
5. Log in to the workstation virtual machine.

CHAPTER 7

CONJUNCTIONS

A conjunction is a word that joins other words, phrases, parts of sentences, or sentences together.

Common Conjunctions:

and

Usually joins two positive or two negative ideas.

but

Used to connect two opposing statements.

or

Provides a choice between ideas.

so

Demonstrates an effect or a result, similar to *then* in an **if/then** statement.

Transforming Conjunctions into Simple Sentences

Conjunctions are an important part of effective writing, but they add complexity to sentence structure. When writing technical documentation, analyze whether a conjunction makes a sentence more or less clear than two separate statements.

This sentence can be understood, eventually, but it's confusing:

Foo2Bar is a framework that makes it easy to create application images, and takes application source code as an input, and produces a new image that runs the assembled application in an isolated sandbox.

See how much clearer it is when written as three separate sentences:

Foo2Bar is a framework that makes it easy to create application images. It takes application source code to produce an executable image. Foo2Bar images run in an isolated sandbox for increased security.

Another example:

To distribute your modifications, gather the configuration files, documentation files, all build materials, including build scripts and scripts needed to dynamically generate makefiles, and compiled binaries, and create a tar archive, or use the **foopkg** command on the parent directory.

This becomes much clearer when rewritten with fewer conjunctions:

There are two ways to distribute your modifications. You can create a tar archive manually or you can use the **foopkg** command. For each, you must include all compiled binaries, configuration files, documentation, build scripts, and scripts to dynamically generate makefiles.

Choosing the Right Conjunctions

Sometimes conjunctions are the right choice, and in those cases you must choose the correct conjunction.

But

The *but* conjunction is very inflexible. It is appropriate when the two statements being connected are exclusive of one another. These two examples have very different meanings:

A car can drive but it cannot fly.

Most importantly, *but* implies that two statements are opposed even if they are not. Using *but* incorrectly confuses readers. For example, if you write:

Linux is open source but it is a stable, Enterprise-ready platform.

A reader probably interprets it to mean that Linux is Enterprise-ready *in spite of* it being open source. What you actually mean is that Linux is *both* open source and Enterprise-ready.

Linux is open source and is a stable, Enterprise-ready platform.

And

The *and* conjunction connects two equally true or equally false statements into one:

That hat is not red and it is not mine.

This fedora is red and it is mine.

Or

The *or* conjunction implies a choice between the two statements it connects:

Run the **DHCP** daemon embedded on the router or run your own on a server.

In the preceding statement, it's significant that you can use either the embedded DHCP server or your own, because using both could cause conflicts severe enough to bring down your network. It would be inaccurate to use a different conjunction:

Run the **DHCP** daemon embedded on the router and run your own on a server.

In fact, for added emphasis, you can use *either*:

Run either the **DHCP** daemon embedded on the router or run your own on a server.

So

The *so* conjunction connects causes and effects:

Red Hat Enterprise Linux 7 is the latest release. This server is running Red Hat Enterprise Linux 6, so it needs to be updated.

This server is not running firewalld or SELinux, so it does not pass my security audit.

Exercise

Practice using the correct conjunction, or rewrite to use no conjunction:

1. Linux is an excellent platform for development, but it is an excellent platform for deployment.
2. Security-enhanced Linux (SELinux) is an implementation of a mandatory access control mechanism in the Linux kernel. It is an important part of system security. Do not disable SELinux, so use the **setenforce** command to put SELinux into permissive mode temporarily.
3. Click the **OK** button to proceed to the next screen, and then click the **Name** link in the upper-left corner of the page or the **Name** selection in the **Navigation** drop-down menu in the upper-right corner of the page, and click the **arrow** icon to continue.

CHAPTER 8

PUNCTUATING SENTENCES CORRECTLY

Punctuation is important because it defines the scope of sentences and clauses. Using punctuation incorrectly can confuse the reader or even change the meaning of a sentence.

Example

In the interest of clarity, this is a simple example:

Let's program everyone.

Taken as written, that sentence means "let us take everyone and program them with code" but probably what the writer intended is "Hey, everybody, let's go write code!"

Period, Full-stop, Dot

The most basic punctuation mark is the *period* or *dot*, sometimes called a *full-stop*. The period signals the end of a complete sentence. It gives the reader a chance to process whatever information was contained in a sentence.

A common misuse of a period is ending a sentence before it is complete. A complete sentence in English requires a subject and a verb in an independent clause.

This is a complete sentence because it has a subject and a verb:

Computers calculate.

This is not a complete sentence because, while it does include a subject and a verb, it is not an independent clause:

Boot the computer and then.

Periods are more than just technical requirements to end a sentence, though. They help simplify complex concepts by breaking them down into small chunks of information that are easier for the reader to process.

This sentence is complex:

Before launching the Foo service, you must ensure that your user is a member of the foo group by opening a terminal and using the **usermod** command to append the group to your user's group membership (see the **usermod** man page for more information) and then logging out so that the new group list is loaded, then start the service with the **systemctl** command or with the included startup script.

It is much easier to understand when broken into smaller sentences:

Before launching the Foo service, you must ensure that your user is a member of the foo group. To add your user to the foo group, open a terminal. In your terminal window, use the **usermod** command to append the group to your user's group membership. If you are unfamiliar with **usermod**, refer to its man page. After you have added your user to the group, you must log out and then log back in so that your group membership is refreshed. To confirm membership, use the **groups** command. To start the service, use the **systemctl** command, or the included startup script.

It is a lot of information, and probably ought to be made formally numbered steps, but replacing commas and conjunctions with periods is an important first step toward simplification.

Comma

The comma is used to separate items in a list, or to separate independent sentence clauses.

Lists

A list is a series of individual words meant to be taken as a group. For clarity, use a comma between each unique item in a list, even the final item.

For this project, you must use Emacs, XML, and xsltproc.

Intentionally omitting a comma implies an association within a list. For example:

For this project, you must use Emacs, XML and the Docbook schema, and FOP.

In this example, not using a comma between *XML* and *the Docbook schema* implies to the reader that they will only be using XML by using Docbook, but won't need a deep understanding of how XML works in general.

Clauses

Commas are also used to separate sentence clauses. A sentence clause is some segment of a complete sentence. A clause might be a complete sentence, or it might be dependent on the rest of the sentence to make sense.

Tip

Dependent clauses are artifacts of complex sentences. If you write with minimalism in mind, you write fewer dependent clauses. If you write fewer dependent clauses, you have fewer commas to worry about.

This sentence consists of two clauses. One is independent and the other dependent:

I use the open source Mumble audio chat application to provide technical support to my users.

To know where to place a comma, you must identify sentence clauses. The easiest way to identify a clause is to find the essential parts of the sentence. In the example, "I use the open source Mumble audio chat application" is a complete sentence on its own. It is an independent clause.

The clause "to provide technical support to my users" is meaningless alone, because it is not a complete sentence.

The sentence can be rewritten with a comma separating the two clauses:

To provide technical support to my users, I use the open source Mumble audio chat application.

The same is true for very small clauses:

POSIX operating systems, such as Linux and BSD, are the only ones that support the nproc resource limit.

The dependent clause "such as Linux and BSD" is surrounded by commas to isolate it from the independent clause it interrupts.

Colon

The colon expands on a sentence by introducing a list. Think of it like a period, but with an addendum, because the sentence a colon finishes is an independent clause.

There are many advantages to Linux: stability, rapid development, open source code, and an innovative, global user base.

Semicolon

Semicolons also introduce a list, but they can also be used like a comma to separate two independent clauses.

Semicolons are usually unnecessary and add complexity to your sentences. If you can, avoid them. They are often indicative that a sentence can be rewritten more elegantly and clearly.

For example, this simple sentence is correct:

You should use Linux; it's a good operating system.

But for it isn't actually as clear as it could be:

You should use Linux because it's a good operating system.

Semicolons are often used to introduce lists:

The Internet relies on several open source technologies; TCP/IP, DHCP, and DNS.

Parentheses, Rounded Brackets

Parentheses serve the same function as commas, with the implication that they are commenting on a sentence without adding to the sentence.

Like semicolons, parentheses are often unnecessary and are best when used as an inline bibliography or reference.

In this example, parentheses are intended to bring clarity to a sentence without interrupting the flow of the sentence:

Before launching the Foo service, you must ensure that your user is a member of the foo group by opening a terminal and using the **usermod** command to append the group to your user's group membership (see the **usermod** man page for more information) and then logging out (so that the new group list is loaded). Then start the service with the **systemctl** command (or with the included startup script).

The paragraph can be rewritten without parentheses for greater clarity:

Before launching the Foo service, you must ensure that your user is a member of the foo group. Open a terminal and use the **usermod** command to append the group to your user's group membership. See the **usermod** man page for more information on the command. Log out and the log back in. Start the service with the **systemctl** command or, if you are not using **systemd**, with the included startup script.

However, as a bibliographic reference, the parentheses are acceptable:

Open a terminal and use the **usermod** command to append the group to your user's group membership (see "Usermod Pocket Guide").

Parentheses are also appropriate when identifying initialisms and acronyms:

Use Alpha Bravo Charlie (ABC) for secure connections.

Question Marks

Question marks are specific to sentences that pose a question. In technical writing, they are usually only used in quizzes or discussions.

You should not pose questions to the reader in an attempt to prompt a silent answer, the way you might in personal conversations:

Incorrect

Before you are able to set up your computer for remote login, you must know its IP address. And how do we find that out? We use the **ip** command, of course!

Correct

Before you are able to configure your computer for remote login, you must know its IP address. Obtain your local IP address with the **ip** command.

Quotation Marks

Quotation marks represent something someone has said, or in some cases, something that people often say.

In technical writing, there are few instances when quotation marks are required outside of configuration files or commands. The typical use case for quotes in technical writing is when you want to refer to an unofficial but widespread term.

The image has been preconfigured, or "pre-baked", with all the options you need for this exercise.

Exercise

Add missing punctuation where necessary.

1. If all of the file's data has been deleted the file can be reclaimed and reused by the broker.
2. This controls the application's behavior on transaction boundaries such as commit, or rollback.
3. One major concern for administrators when considering data integrity of a broker's persistent data is whether or not the disk that is backing the broker is using a write cache.
4. The broker creates as many files as needed but when it is reclaiming space it shrinks the number of journal files back to the `journal-pool=files` number.
5. Notice that this time the error message is made available to the application, even after foo was restarted.
6. Use the **mysql** command-line interface to log in and select the log data from the **foo** database.

CHAPTER 9

USING THE CORRECT POSSESSIVE

The possessive form indicates ownership. Nouns and pronouns have possessive forms.

Converting Nouns into Possessive Nouns

These are the different rules for converting nouns into their possessive forms:

Add 's to singular nouns:

An *administrators's* computer.

A software *project's* blog.

Add a lone ' to the end of a plural noun that already ends in s:

Update all of the *students'* computers.

Email each of the *projects'* reports to the project manager.

Treat plural nouns that do not end in s the same as a singular noun:

Install Linux on all of the *children's* computers.

Bring those *people's* laptops in for repair.

If a noun is compound, convert only the final noun to its possessive form:

Install Linux on *admin and manager's* computers.

Bring the *student and teachers'* laptops in for repair.

Converting Pronouns into Possessive Pronouns

Pronouns have their own special possessive forms, separate from their root words. In technical writing, the gender neutral ones are the most common:

For personal ownership in the third person, use *their*, whether it refers to a singular or plural noun:

Give your user a GPG key so that their email can be encrypted.

Give all of your users GPG keys so that their emails can be encrypted.

For personal ownership in the second person, use *your*:

Do not give your password to anyone.

If a pronoun refers to a nonhuman, use *its*:

Post the checksum of an RPM so its integrity is verifiable by users.

The plural form of *its* is also *their*:

Post the checksums of all your RPM packages so that their integrity is verifiable.

Using *Its* and *It's*

The words *its* and *it's* notoriously get confused for one another in writing. You don't think about the words when you speak, because they're pronounced exactly the same way, but when you write them, it matters which one you use.

There are two simple rules you can use to remember which to use:

1. The word *it's*, like all contractions, is a combination of two words: *it* and *is*. Whenever you see *it's* in a sentence, replace it with *it is*. If it does not make sense, then it is not used correctly. For example:

Open the computer and re-seat *it's* RAM.

Open the computer and re-seat *it is* RAM.

In this example, it makes no sense to say "re-seat *it is* RAM." Obviously, the writer intended to use *its*.

2. In technical documentation, contractions are generally avoided. Use *its* as a possessive pronoun, and use *it is* instead of *it's*.

The following **grep** statement checks for *it's* in a file:

```
& grep -iE "it's" filename.xml
```

Using *Your* and *You're*

The words *your* and *you're* also get confused for one another in writing.

The same rules about *its* and *it's* apply to *your* and *you're*. To reiterate:

1. The word *you're*, like all contractions, is a combination of two words: *you* and *are*. Whenever you see *you're* in a sentence, replace it with *you are*. If it does not make sense, then it is not used correctly:

To get the most out of a Linux desktop, replace *you're* single-button mouse with a three-button mouse.

To get the most out of a Linux desktop, replace *you are* single-button mouse with a three-button mouse.

In this example, it makes no sense to say "you are single-button mouse." Obviously, the writer intended to use *your*.

2. In technical documentation, contractions are generally avoided. Use *your* as a possessive pronoun, and use *you are* instead of *you're*.

The following **grep** statement checks for *you're* in a file:

```
& grep -iE "you're" filename.xml
```

Exercise

Rewrite these sentences, as needed, using the correct possessive forms:

1. Press the **Enter** key to convert the file and view it on its screen.
2. Linux and BSD both use monolithic kernels as its basis.
3. The library's and binary's maintainer forgot to upload the packages.
4. Have each user activate two-factor authentication on his account.
5. You must use **mutt** as the email client.

CHAPTER 10

LISTS AND TABLES

There is no single standard for writing a list. The style of each list depends on the contents of the list, but the style should remain consistent within each list.

There are, broadly, three different types of lists: the ordered list, the itemized list, and the variable list. When a reader must follow the items in a list sequentially, use an ordered list. If sequence is not important, use an itemized list. When defining a list of terms, use the variable list.

Stems

Avoid using bullet point lists to format a single sentence. Some translation tools, for example Zanata, display list items and the introductory sentence (or *sentence stem*) as individual sentences for translation. If these are not complete sentences, they are difficult to translate.

Example	Improvement
Before you start the installation, make sure you have <ul style="list-style-type: none">• enough free storage on your system• backed up any data that you want to keep to ensure a smooth installation.	Before you start the installation, follow these steps to ensure a smooth installation: <ul style="list-style-type: none">• Ensure you have enough free storage on your system.• Back up any data that you want to keep.

Punctuation

If you place a period at the end of one list item in a list, then use a period at the end of each list item in that list.

Use punctuation when a list item completes an introductory sentence.

Before removing your USB hard drive from your computer, you may need to:

- close any open files located on the drive.
- close any terminal open to a directory on the drive.

Use punctuation when a list item contains more than one complete sentence.

Before removing your USB hard drive from your computer, there are two tasks you may need to take.

- If you are running an application with a files from the drive open, you must close the file. Alternately, save a new copy of the file to some other location. If you are not sure whether you have any files from the drive open, use the **ls -o** command.
- If you have a terminal open to a directory on the drive, either close the terminal or change directory to a directory not located on the drive.

Do not use punctuation for:

- sentence fragments
- single words or short terms

Most of all, strive for consistency. If a list contains two list items that are complete sentences, and a third that is just a single word, you should restructure your list. For example, change the single word list item into a simple sentence so that it matches the other items. Alternately, review whether the items in your list belong in a list together.

Tense

Retain the same verb tense for each list item.

For example, this is incorrect:

- Students gain an understanding of the Linux init system.
- All students will learn how to use **systemctl**.
- Students create a simple chroot for testing.

The second list item uses the future tense, while the other two are present tense.

This is correct:

- Gain an understanding of the Linux init system.
- Learn how to use the **systemctl** command to start, stop, and monitor services.
- Create a simple chroot for testing.

This is called *parallelism*, and should be applied across all lists within the same book.

A list that constantly changes sentence structure is confusing:

- Use to specify the X variable.
- Specifies which value to apply to Y.
- You can use this to change the config from a POSIX shell.

It is easier to understand when a list maintains the same voice throughout. A reader might refer to several lists or tables within a book, so ensuring a consistent voice in every list or table helps readers quickly parse important information.

Key-Value Lists

If you are writing a list that exists only to define terms, use a variable list.

`example.com`

A domain reserved as a safe example URL. This is the correct way to define a term in a list.

`example.local`

A domain that exists only within a local network.

`cat`

The concatenate command.

Sometimes your list definitions do not contain exclusively simple terms to be defined, so you have to use an itemized list to allow for mixed list items. Delimiters separating a term from its definition can be confused as being a part of the term, so it is best to define terms in complete sentences, and with appropriate Docbook tags.

- `example.com` is a domain reserved as a safe example URL. This is the correct way to define a term in a list.
- `example.local`: a domain that exists only within a local network. This is not as clean, since the colon (:) could be interpreted as part of the term being defined.
- **`cat`** - the concatenate command. This is also unclear, since the dash (-) could represent shorthand for POSIX `stdin`.

Tables

Tables are, essentially, lists rendered horizontally as well as vertically.

The same rules apply for tables as for lists, with the primary goal being internal consistency. This table is not consistent:

Server	OS	Function
servera	RHEL 7	DNS
serverb	RHEL 7	DNS fallback
serverc	Fedora 26	This is only a testing server.

This is a consistent table:

Server	OS	Function
servera	RHEL 7	DNS
serverb	RHEL 7	DNS fallback
serverc	Fedora 26	This is only a testing server

Even though the final entry in the table is a complete sentence, it has no period at the end of the sentence to better match the convention established by previous entries.

CHAPTER 11

CREATING REFERENCE SECTIONS

Referencing internal and external content is common in technical documentation. To avoid linking to dead links or outdated content, you must create reference links carefully.

Referencing Red Hat Docs within an Admonition

Red Hat documentation published on docs.redhat.com does not have permanent URLs with chapter and section titles. For this reason, do not imply that you are directly quoting a chapter title in your reference, but give some indication of the chapter or topic that the reader needs to look for.

```
<note role="References"> <para> Additional information may be available in the
chapter
on Kickstart installation in the
<emphasis>Red Hat Enterprise Linux Installation Guide</emphasis> for Red&nbsp;Hat
Enterprise&nbsp;Linux&nbsp;7, which can be found at <ulink
url="https://access.redhat.com/documentation/en-US/index.html"/>
</para></note>
```

References

Additional information may be available in the chapter on Kickstart installation in the *Red Hat Enterprise Linux Installation Guide* for Red Hat Enterprise Linux 7, which can be found at

| <https://access.redhat.com/documentation/en-US/index.html>

Generic Referencing within an Admonition

For links pointing to documentation external of Red Hat, link to the documentation directly as possible. Use the `<ulink>` tag instead of the `<uri>` tag if you are including a URL in a reference.

```
<note role="References">
  <para>
<ulink url="http://docs.fedoraproject.org/en-US/
  Fedora_Contributor_Documentation/1/html/Users_Guide/">Publican Users Guide</ulink>
  </para>
</note>
```

References

Publican Users Guide
[http://docs.fedoraproject.org/en-US/ Fedora_Contributor_Documentation/1/html/ Users_Guide/](http://docs.fedoraproject.org/en-US/Fedora_Contributor_Documentation/1/html/Users_Guide/)

Combining Multiple Reference Types

When combining more than one reference type, list them in one reference note at the end of a section.

There are many sources you might use as reference. According to the type of material, place them in the following order of priority:

1. Info nodes
2. man pages
3. yelp references (RARE)
4. Knowledgebase or Customer Portal references
5. Third-party URLs
6. Red Hat Documentation

Info nodes

Info nodes are referenced first.

```
<note role="References">
<para>
  <command>info libc</command> (GNU C Library Reference Manual)
  <itemizedlist>
    <listitem>
      <para>Section 29.2: The Persona of a Process</para>
    </listitem>
  </itemizedlist>
</para>
</note>
```

References

info libc (*GNU C Library Reference Manual*)

- Section 29.2: The Persona of a Process

man pages

Man pages should be listed as a single comma-separated list of items. The order of the man pages is at your discretion. The section of the manual for the man page must always be listed in parentheses after the man page title, even if there's no man page of the same name in any other section of the online manual. The section number must be outside the `<command>` element.

```
<note role="References">
  <para>
    <command>ls</command>(1), <command>mv</command>(1), and <command>su</command>(1)
    man pages
  </para>
</note>
```



```
</para>
</note>
```

References

ls(1), **mv**(1), and **su**(1) man pages

Yelp (GNOME help)

When referencing a page from GNOME help, try to reference a command that can be used to bring up the exact page. Deep links to individual pages can be determined by looking at the ***.page** files in **/usr/share/help/C** in the **gnome-help** directory.

```
<note role="References"> <para> GNOME Help: <emphasis>Getting Started with GNOME</emphasis>
  <itemizedlist>
    <listitem>
      <para>
        <command>yelp help:gnome-help/getting-started</command>
      </para>
    </listitem>
  </itemizedlist>
</para>
</note>
```

References

GNOME Help: *Getting Started with GNOME*

- **yelp help:gnome-help/getting-started**

Local documentation

URLs pointing to documentation on web pages served by the local machine. This is rare, but two significant examples are the CUPS help pages and the httpd-manual pages.

```
<note role="References">
<para>
  <ulink url="http://localhost:631/help/">CUPS Online Help</ulink>
</para>
</note>
```

References

CUPS Online Help
<http://localhost:631/help/>

Knowledgebase

Knowledgebase articles are stable. Reference them with a `<ulink>` directly to the article. Titles should be verified. If the article title doesn't match and seems wildly different, the author must validate that the article reference is correct.

```
<note role="References">
<para>
  <ulink url="https://access.redhat.com/site/solutions/253273"> Knowledgebase:
    &quot;How to register and subscribe a system using Red Hat Subscription Manager&quot;;
  </ulink>
</para>
</note>
```

References

Knowledgebase: "How to register and subscribe a system using Red Hat Subscription Manager"
<https://access.redhat.com/site/solutions/253273>

External docs

External (non-Red Hat) URL references go last. These should be used sparingly, and only when the site is truly an authority.

```
<note role="References"> <para>
<ulink url="http://www.rfc-editor.org/rfc/rfc2460.txt"> RFC 2460: Internet
Protocol,
Version 6 (IPv6) Specification</ulink>
</para>
<para>
<ulink url="http://www.ntp.org/">NTP: Network Time Protocol</ulink>
</para>
<para>
<ulink url="http://www.pool.ntp.org/en/use.html">NTP Pool Project: How do I use
pool.ntp.org?</ulink>
</para></note>
```

References

RFC 2460: Internet Protocol, Version 6 (IPv6) Specification
<http://www.rfc-editor.org/rfc/rfc2460.txt>

NTP: Network Time Protocol
<http://www.ntp.org/>

NTP Pool Project: How do I use pool.ntp.org?
<http://www.pool.ntp.org/en/use.html>

APPENDIX A

REVISION HISTORY

Revision Thu Mar 9 2017
0.0-0

Seth Kenlon ***skenlon@redhat.com***

Initial creation by publican

INDEX

F

feedback1

contact information for this brand, xi
