English for Techies

English Lessons for Tech Writers

Seth Kenlon , **Red Hat** `<skenlon@redhat.com>`
David O'Brien , **Red Hat** `<daobrien@redhat.com>`

English for Techies: English Lessons for Tech Writers

by Seth Kenlon and David O'Brien

Abstract

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.

Table of Contents

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!

You should over ride this by creating your own local Feedback.xml file.

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 dialogue 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 dialogue 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 tellying 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. 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 a 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 4. 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 5. 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 phase 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 6. Conjunctions

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

Common conjunctions are:

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 7. Using the right 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 all ready 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.

Appendix A. Revision History

Revision History
Revision 0.0-0   Thu Mar 9 2017   Seth   Kenlon
< skenlon@redhat.com >

Initial creation by publican

Index