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 iv
Document Conventions iv
Typographic Conventions iv
Pull-quote Conventions v
Notes and Warnings vi
We Need Feedback! vi
1. Minimalism 1
How to implement minimalism 1
Exercise 2
2. Active Voice 3
How to write in the active voice 3
Exercise 4
3. Definite and indefinite articles 5
Exercise 5
4. Pronouns 6
How to use pronouns 6
Exercise 7
A. Revision History 8
Index 9

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. Minimalism

"Minimalism", in the context of technical writing, is writing less to write clearly.

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.

How to implement 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. 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 an exercise requires it. For technical writing, the active voice is the clearest way to demonstrate action and result, or cause and effect.

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

Active voice. I compiled the code.

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:

Passive voice. 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.

For instance, it is impossible for a reader to know with certainty whether the sentence "A key being pressed prompts the computer to continue," is an instruction to press a key, or 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."

How to write in the active voice

Use the active voice by writing 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 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 paragraphs to minimalism

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 PyFoo code base.
Active voice:	The syntax for XML configuration, and the steps to import the required module into PyFoo, were explained in the *Importing the BeautifulSoup module* section.

Exercise

Rewrite each paragraph in the active voice.

1. The QTextEdit field should be placed in the right corner of the QMainWindow.
2. To begin the code compilation process, GCC is used.
3. The previous three steps should be repeated to complete RHSM registration on each machine.

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:

  Use the `for` loop from the code sample to iterate over each file.

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, it's common to use generic terms, such as a job title instead of an arbitrary name, such as "John" or "Jane", and 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 RHEL 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 for a small network.

Possessive pronouns. To indicate that a pronoun possesses something, use a 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.

Pronouns in context. Sometimes a pronoun is used in a separate sentence from the one that establishes the noun it references. This relies on the reader to understand from context which noun the pronoun most likely refers to.

Sometimes it is very obvious. In the phrase "I like drinking coffee while I sit and type at my computer. Sometimes, I even dunk a doughnut into it," the reader knows from contextual clues that the writer means "Sometimes, I even dunk a doughnut into my coffee" and not "Sometimes, I even dunk a doughnut into my computer."

Sometimes it is less obvious. In the phrase "The foo command is working to replace the bar command. It is installed by default, and it has a bug in it that erases all of your data permanently, so must not be used," the reader cannot know definitively which command erases all of their data and which command is safe. Sometimes, using a pronoun is not as effective as being explicit.

Verify your 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 desktop computers can also be configured as servers for a small network.

If it still says what you meant to say, then you have placed the pronoun in the correct place. 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.

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.

Appendix A. Revision History

Revision History		
Revision 0.0-0	Thu Mar 9 2017	Enter your first name here.Enter your surname here.<Enter your email address here.>

Initial creation by publican

Index

F

feedback
    contact information for this manual, vi