

# Designing and Writing Procedures

David K. Farkas, Professor Emeritus  
Department of Human Centered Design & Engineering  
University of Washington, Seattle, WA, USA

## Introduction

“Procedures” explain how to do something. Another word for procedures is “instructions.” Procedures are all around us, for example:

- Printed product instructions
- Online help systems for software products (consist *primarily* of procedures)
- Short video tutorials posted on eHow.com and elsewhere
- Recipes
- Face-to-face instructions of all kinds (“When the road forks, turn right.”)

The main components of typical procedures are the title, an introductory paragraph, and a list of numbered steps. But as you will see procedures are more complex than this.

We need to distinguish between procedures and closely related forms of discourse:

- **Process descriptions** are similar to procedures but explain something that happens without a human agent, such as photosynthesis.
- **Product demos** (often videos) resemble procedures and may even show how to use the product. But their main purpose is to sell a product by explaining its features and benefits.
- **Functional descriptions** make clear what something does but assume the user knows what action to take. If the phrase “Coolant pressure relief” is stenciled near the hand wheel of a gate valve in an industrial setting, it is assumed that the workers know what action they need to perform if they need to relieve the coolant pressure. Similarly, a tooltip may read “Increase font size” (the purpose of this toolbar button), but there is no explicit “how to” information.

Professionals in many fields create procedures and are responsible for products and services that will not be successful if these procedures are not usable. A knowledge of procedure writing, therefore, is a valuable skill for a wide range of professionals as well as technical communication professionals.

Procedures are not especially hard to write but require some skill. We often encounter bad procedures. Some of the causes are these:

- Unskilled writers. One common problem is that the writer is not able to adapt to the audience. This leads to various problems, most often procedures that are too technical for the audience or that leave out important information that the writer forgot to include.
- Cost cutting. Some consumer products provide barely legible instructions that are squeezed onto a single sheet of paper devoted mainly to legal warnings. Often manufacturers distribute one set of instructions for different models of a product, and so users struggle to figure out what information applies to the model they purchased. To minimize translation costs, manufacturers try to get away with “near zero text” procedures that consist almost entirely of graphics. This approach may work for the assembly of a simple bookshelf, but it rarely works for more complex products. Sometimes procedures are poorly translated into the language of the target audience.

## Key concepts for understanding and writing procedures

Here are some key concepts for understanding procedures.

1. The **domain** of a procedure
2. The **system state**: Does the procedure deal with the assembly, operation, maintenance, or repair of a system?
3. The **audience** (“users”): Especially the audience’s particular backgrounds and information needs, but also general characteristics that are very widely shared among human beings
4. The **modality**: In particular, whether the procedure employs text, text with graphics, or video

### *The domain*

The domain is the kind of system being explained. Much procedural discourse is written to explain how to use **digital systems** (such as software applications, websites, or smart phones). Although procedures for digital systems may be conceptually difficult (“Configuring Glype on your proxy server”), digital systems employ a carefully crafted user interface (GUI) that greatly simplifies the task of writing the steps. Even if a manual or help system contains a hundred or more procedures, the user repeatedly performs very similar operations (clicking, dragging, scrolling, etc.) on the same few types of GUI elements: tabs, menus, dialog boxes, option buttons, etc.

**Mechanical and electrical equipment** require the user to interact with such physical components as switches, latches, and so forth (although such systems may also include a GUI display). Various complications arise, such as explaining when a part is worn out and how tight to fasten something.

Explaining how to carry out tasks in the **natural world** is often difficult. Consider, for example, instructions for shucking an oyster. An oyster was not designed to be opened up by an knife.

There is no user interface. It is quite challenging—even with video—to make clear how to grasp the oyster, how to find the part of the shell that covers the hinge, where to insert the knife point, and how to work the knife with the correct twisting motion. Similarly, it is challenging to create instructions for serving a tennis ball or hitting a golf ball.

There are also procedures (often in book form) for **intangible subject** areas such as leadership, running a meeting, or remembering the names of people you meet.

### *The system state*

Most systems exist in more than one state. So, for example, we write procedures for getting a system going (assembly, installation, configuration), for operating the system (routine and emergency operation), for maintaining the system, and for repair. Often when systems malfunction, the problem is not obvious, and so troubleshooting procedures are often necessary. Troubleshooting procedures first guide users in identifying the problem and then provide one or more solutions for that problem. For an introduction to troubleshooting procedures (focused on digital systems), see

<http://faculty.washington.edu/farkas/dfpubs/Farkas-TroubleshootingProcedures-QuikScanned.pdf>

### *The audience*

As noted above, understanding and adapting to the backgrounds and information needs of the audience is central to writing procedures, as it is to all forms of communication. The background of the intended audience, in particular their **level of experience and expertise**, in large part governs their information needs—what terms and concepts the writer needs to explain and the pacing of the procedure. That is, for more advanced audiences the pacing can be faster and you can write higher-level (and therefore fewer) steps than are necessary for entry-level (or “novice”) audiences. A very difficult problem in writing procedural discourse is how to write for audiences with very different backgrounds.

One important dimension of an audience’s background is **culture**. Instructions for products sold worldwide require metric (SI) as well as English units of measurement. Also, if a manual or video shows people using a product, you should take care that their clothing, behavior, and other elements of the visual do not offend or confuse people from certain cultures.

An important dimension of information needs is whether to design for **retention**. For example, procedures in first-aid manuals are optimized for fast reading and immediate action. But for *training* EMTs (emergency medical technicians) to provide first aid (with minimal or no recourse to procedures) very different procedures are prepared. These training procedures include previews, reviews, quizzes, and other elements that promote retention. Note also that sets of first-aid procedures are routinely classified as members of the genre “first-aid manual.”

However, because first-aid procedures usually begin with diagnosis (Is the person breathing? Are his lips blue?) they can be classified conceptually as repair/troubleshooting procedures for a natural system, the human body.

When we think about audience, we usually think about differences. But we can also consider characteristics that are essentially universal (based on hard-wired aspects of human perception and information processing) or are at least shared across a very broad range of user categories, including users with different cultural backgrounds. All human beings have **limitations to their short-term memories**, and these limitations fall within a well-established range, with few outliers. Therefore, in all our design and writing decisions we must be cognizant of the cognitive load we are creating, and we must strive to reduce this cognitive load.

Another important general characteristic of users is that they benefit from and typically appreciate **well-designed examples**. Because examples are concrete and (as much as possible) draw upon the users' previous experience, they can reduce cognitive load. You can embed a very brief example in a single numbered step. Tutorial documentation employs one or more lengthy examples, sometimes called "scenarios," as the narrative framework into which multiple procedures, along with components for retention, are embedded.

Furthermore, all human beings have competing demands on their time and are often **impatient** reading procedures—especially product instructions—for which the time required seems too great for the benefits expected. Users, therefore, are tempted to bypass instructions even though this decision is very often counterproductive—they would ultimately have done better reading the instructions. Are you an impatient reader of procedures?

Because of this general human characteristic, we must strive to make procedures fast and easy to work with, make them visually appealing, and perhaps find ways to make them interesting and enjoyable. Furthermore, it is often a good strategy to subtly "sell" our procedures. Conversely, procedures with a confusing visual design and poor writing not only increase the user's cognitive load but immediately lose credibility and make users more apt to set them aside.

## Modality

Modalities are symbol systems through which information is encoded. Voice, text, visuals, video, and sign language are all modalities.

### *Text and visuals*

Most procedures communicate primarily through text and visuals. Visuals can be purely conceptual (for example, a flowchart). Very often, however, visuals show the appearance of a device (or parts of the device) or its user interface, and they indicate where and possibly how to carry out an action. They can also show what the system should look like when an action has been carried out successfully (feedback).

There are many considerations in the design of visuals. Photographs are easy to create and show very realistically what an object looks like. However drawings, in particular line drawings,

provide a more filtered view: the artist can show only what is important for the purpose of the visual. Consider, for example, the potential problem of unwanted shadows in a photograph. Yes, it is now possible to convert photographs into line drawings, but the basic distinction still holds.

Another issue is the “view.” Are we looking at the device from the top or side (and which side is best)? Do we want an “exploded view”? How much of the device should be included? A broad view often makes it harder for the viewer to locate the element of interest within the visual. But if the view is too narrow (just a close-up), the viewer loses context and may not be able to locate the element represented in the visual on the actual device. Still another consideration is the coordination of the visual with the text, including the effective use of captions and callouts.

### *Video*

The web and other digital media allow the use of narrated videos—which we divide into (a) photorealistic videos (recorded with a video camera), (b) drawn animations, and (c) captures of activity on a computer screen. If we look back to the example of shucking an oyster or swinging a golf club, we see that photorealistic videos and drawn animations are very beneficial for documenting tasks that (1) take place in the natural world and (2) involve motion. Drawn animations have advantages over photorealistic videos similar to the advantages of drawn graphics to photographs. The ideal video for explaining how to shuck an oyster would consist of both photorealistic segments and segments with drawn animation.

Screen-capture videos are often used to document digital systems. Adobe, for example, has created libraries of video procedures to teach Photoshop, Dreamweaver, and other products. When such video procedures are well designed, users benefit from receiving information from both their visual and auditory channels. Because the visual channel is very good at showing where GUI elements are, how to work with them, and what the result is (feedback), the narration can focus on the purpose of these actions, which is hard to communicate visually. Even so, we should not assume that video procedures are clearly superior to print procedures, especially because users do not have complete control of videos (and other time-based media) as they do with text and static graphics. You can easily make your own video computer procedures (akin to the Adobe video procedures) with online screen-capture services such as Screencast-O-Matic.

### *Voice only*

We should not forget voice-only instructions. In both our non-work and professional lives we are often called upon to give oral instructions (although we can often grab a pencil or even “draw in the air” with our fingers). Furthermore, in the automobile industry, audio-only procedures that include an interactive dialog with the driver are becoming prevalent.

### *Modality vs. medium (A bit of theory for folks who are interested)*

In the preceding paragraphs voice, text, visuals, and video could have been termed “media” (as is often done), but this would be technically incorrect. A medium is actually the technology

through which content is communicated through space and time (along with the business model and other societal arrangements that are part of each technology). Print, therefore, is a medium (or cluster of media), as is the web. Modalities are symbol systems through which information is encoded. Particular modalities are possible or not possible in particular media. Text, for example, is a modality that is possible in print, on the web, and even in cinema (for example, the scrolling text used for credits) but is not possible in traditional telephony. Media do more than just convey certain modalities. For example, the digital media permit powerful behaviors such as the display and hiding of blocks of content, hyperlinking, and digital search (vs. the print index).

## How to write procedures

Here is a very practical approach to writing procedures in which procedures are treated as a set of mandatory and optional components, each serving a particular function. The figure provides an overview. Unsurprisingly, the construction of procedures varies enormously, and you will encounter procedures that are structured and phrased differently from those described here. But even so, these next few pages can take you a long way toward being a first-rate writer of procedures.

**Import Your Music** Topic title

The Tuner Music Importer helps you add your personal music collection to your music library on the Tuner Music website, so you can play it from all of your connected Tuner Music devices.

Introductory paragraph

---

1. Go to [your music library](#) from a web browser on the computer you want to import music from.
2. Click **Upload your music** in the left menu. If prompted to install the Tuner Music Importer, follow the on-screen instructions.
3. Once you've opened or installed the Tuner Music Importer, click **Start Scan** to automatically scan your iTunes and Windows Media Player libraries for songs to import. You can also click **Browse manually** to choose files on your computer to find music. This process may take some time to complete.
4. After the Tuner Music Importer has located your music, click **Import all** to add the entire selection to your music library. Or, click **Select Music** to choose which songs you'd like to add.

**Note:** Note

- You can import up to 250 songs to your music library for free. With a Tuner Music subscription, you can import up to 250,000 songs. To learn more, go to [Change Your Tuner Music Subscription](#). Tuner Digital Music purchases do not count towards library limits.

Feedback statement

*A typical procedure showing the most widely used components.*

## The title and introductory paragraph

**The title** identifies the procedure but also very briefly explains its purpose. When a simple consumer product such as a bookshelf ships with a single assembly procedure, the title can be very general—perhaps “Instructions” or “Assembly” or “How to assemble your Ikea X-15 bookshelf.” But if the procedure is part of a large set of procedures, such as the many help “topics” that make up an online help system, the titles must be more specific and informative so that users can choose the procedure they want when numerous titles appear together in a table of contents or some kind of picklist.

The title is often followed by an **introductory paragraph** (or paragraphs). This paragraph provides various kinds of information that enable the user to decide whether to carry out the procedure and/or make it easier for the user if she does so. Often there is more detailed information about the purpose of the procedure than the title can provide so the user can confirm that this procedure does indeed meet her goals. This includes stating some secondary consequence or “side effect” that some users may not want. For example, if a user saves a document created in a word processing application as a .txt (ASCII) file, the file size is much reduced and there are other benefits as well. However, because some users may not know (or may have forgotten) that all formatting will be lost, the introductory paragraph of a procedure entitled “Saving your work as a .txt file” should warn the user about losing all formatting. (The application should also display a system message with a similar warning). As noted above, the introductory paragraph may also include information that makes it easier to carry out the procedure, such as a list of tools or ingredients or background knowledge needed to carry out the procedure (pre-requisites). Of all the procedure components, the introductory paragraph is the one that requires the writer to most deeply analyze the needs of the user.

## Steps and feedback

Following the title and the introductory paragraph (if present), most procedures transition to a list of **numbered steps**, written as commands. The primary role of the steps is to enable the user to execute the procedure she has by now decided on. We can therefore say that the title and introduction are about decision-making and that steps are mainly about executing decisions that have already been made.

Procedures can be written in paragraph form with descriptive (indicative) verbs instead of numbered steps with command verbs. *De Re Metallica*, a lengthy book on how to mine and smelt metal, published in 1556, is written this way: “The tiny flakes . . . of silver adhering to stones or marble or rocks . . . should be smelted in the furnace of which the tap-hole is only closed for a short time.” But this kind of writing and formatting places an extra cognitive load on users and so is not often used today.

There are four different kinds of steps, each with a different function. These are (1) “Do it” steps (the simplest kind of step), (2) user option steps, (3) conditional steps, and (4) local purpose steps. There are also feedback statements that directly follow a step.

A Do it step directs the user to carry out a particular action: “Press the **On** button.” But Do it steps may also state (or show graphically) the location of the object to be acted upon. In software documentation many steps begin with location information (“On the **Home** tab, in the **Styles** group, click . . .”). It may also be necessary to make clear *how* to carry out the step, especially in the domain of mechanical devices: “Release the catch mechanism by pushing firmly at the ends of the latches.” A variation on the Do it step is a step that tells the user not to do something or warns against a bad way to do it: “Do not tighten the bolt excessively or the porcelain may crack.”

Almost any kind of step can be supplemented by a **feedback statement**. Feedback statements describe the result of carrying out an action: “Tap any note in your document. Blue boxes appear around all the notes.” Feedback statements assure the user that she has done the right thing and that the system is responding properly. The feedback statement in the figure assures the reader that the system is responding properly despite the slow response the user may experience. Feedback statements are often unnecessary, and they make procedures longer and more tedious to users. Writers, therefore, must decide (ideally through user testing) which steps really require feedback statements.

In general, steps guide users in carrying out an action they decided to perform after reading the title and introductory paragraph. They do not ask users to make new decisions. **User option steps**, however, present the user with a decision to switch to a variation on the main goal of the procedure. For instance, “If you prefer a crispy-crust pizza, do not use a baking sheet and bake the pizza at 375 degrees.” It would be entirely possible—but very inefficient—to write two separate procedures, one for soft-crust and one for crispy-crust pizza. User options steps, then, while they make individual procedures a little longer and more complex, make a complete instruction booklet, help system, or manual much shorter.

Sometimes the writer knows that users may face an impediment when trying to carry out a step. **Conditional steps** explain how to identify an unwanted condition and address the problem: “If you are in the Transactions view, switch to the Report view.” A conditional step is *not* a user option. If the user is indeed in the Transactions view of this software product, the user will not be able to carry out any procedure pertaining to reports unless the user addresses the condition and makes the switch. Conditional steps can also occur within feedback statements: “If red triangles appear in the window, adjust the positioning knob until black triangles appear.” The need for numerous conditional steps in a set of procedures suggests either that the system is prone to malfunction or that its user interface is poorly designed.

The last kind of step we will discuss is the **local purpose step**. In most cases, a local purpose step is built upon a Do It step but adds some extra words that explain *why* the step is necessary:



“During prolonged use on very dusty surfaces, rap periodically on the filter housing to clear it of dust and ensure maximum suction.” By explaining the purpose of an action, we increase the likelihood that the user will carry out the step. Other local purpose steps are like milestones in a long procedure. Functioning somewhat like feedback statements, they sum up for the user how much she has completed: “Click **Bottom**, to complete the box.”

## Cautions, notes, and tips

Cautions, notes, and tips call special attention to an item of information. Cautions and warnings warn about actions that might harm human beings, damage systems, or result in unexpected loss of data. Many cautions and warnings are conditionals: “If the temperature indicator reaches the red zone, shut down the system immediately.”

Cautions and warnings must be highly visible and must communicate very clearly to all segments of the audience. In addition, they must appear before the place in the procedure where the user can get into trouble. Often the visual design, text message, and even the placement on the product are specified by standards organizations such as ISO (International Standards Organization).

Many kinds of information can be treated as notes, and some notes are specifically identified as tips or hints. Notes can appear almost any place in a procedure, and writers should choose the most useful location. Avoid overusing notes and—especially—avoid placing multiple notes at the end of a procedure. Often the information in such notes really belongs in a step or feedback statement.

## Understanding how the product is used

While it is essential for a writer to know how procedures are constructed, writers also need a detailed understanding of how the product is used and, especially, what aspects of the product are apt to create problems for users. When writers lack this knowledge (or if they are unskilled or unmotivated), their procedures tend to cover what is easy and obvious but skimp on the content users really need. The writing may be fluent and the formatting may be attractive, but users will soon learn that they have been given low-quality content. Don't write these “brain dead” procedures!