

---

## Writing Tasks, Procedures, and Steps

The purpose of many technical documents is to explain how to use a product to accomplish specific tasks. In technical documentation, detailed instructions about how to accomplish tasks are often provided in the form of procedures and steps.

This chapter contains the following sections:

- [Section 9.1, "Understanding the Relationships Among Tasks, Procedures, and Steps"](#)
- [Section 9.2, "Developing Task Information"](#)
- [Section 9.3, "Writing Procedures"](#)
- [Section 9.4, "Writing Steps"](#)
- [Section 9.5, "Checking for Structural Problems"](#)

### 9.1 Understanding the Relationships Among Tasks, Procedures, and Steps

This chapter uses these task-related terms:

- **Task:** Specific work that can be performed. A task includes instructions for completing the work. A task can also include an explanation of why the work might be performed and any prerequisites and examples.

A task can be short and simple, even just one action to complete. A task can also be long and complex. A long, complex task might need to be separated into subtasks to make the task easier to understand.

- **Subtask:** A small, short component of a larger task. A subtask might be one action or one set of actions to complete. A subtask can include prerequisites and examples.

To complete a task, a user might have to complete or choose from multiple subtasks. Some subtasks might be optional or conditional. These tasks might not always have to be completed, depending on the user's situation or the intended outcome. Remember to clearly identify any optional or conditional subtasks.

- **Procedure:** One step or an ordered set of steps that explains how to accomplish a task or subtask.

A procedure can be optional or conditional. A procedure also can include prerequisites and examples.

- **Step:** An instruction that explains how to perform a task or part of a procedure. A short, simple procedure might require only one step. Two or more steps are ordered and are numbered to show the sequence of actions.

A step can be optional or conditional. A step can also include prerequisites and examples.

A short task, such as backing up a system, might require performing one simple procedure. In that case, you might not use all of the guidelines in this chapter.

## 9.2 Developing Task Information

A task is work that is performed for a particular purpose.

Focusing on the task is as important for a developer's guide as it is for a user's guide or an administrator's guide. When identifying tasks, do not become distracted by the interface. The interface, whether it is a browser interface, graphical user interface, application programming interface, or command-line interface, is the means by which a user accomplishes tasks. Your primary focus should be on how to accomplish the tasks, not on how the interface works.

### 9.2.1 Providing Only Necessary Task Information

When you write a task, provide only the information that is necessary to complete the task. In particular, limit an overview to information without which the user cannot complete the task.

To provide just the relevant information, consider including the following in a task:

- An explanation of what the task is
- The reasons why the user needs to perform the task
- Prerequisites for performing the task
- Instructions about how to perform the task
- Examples that illustrate how to perform the task

For most tasks, the instructions about how to complete the task are in the form of a procedure. For information about writing procedures, see [Section 9.3](#).

### 9.2.2 Including Prerequisites

Include any prerequisites that users must consider before performing a task. The risk of users performing an action out of sequence is particularly high with online documentation because users can enter a task from various points.

For a task that is written as a procedure, follow these guidelines:

- If the task contains information that users must know before performing the procedure, then include that information in an introductory paragraph.
- If users must perform a prerequisite step or procedure, then make the first step the prerequisite step or add a cross-reference to the prerequisite procedure.

### 9.2.3 Providing Examples

Consider including one or more examples whenever doing so can help the user. Do not provide an example if the task is self-evident.

When providing examples, follow these guidelines:

- If the example requires clarification, then include text with the example.
- Keep the example short, showing only the necessary elements.

If the output is lengthy or used only for verification, then show just the first lines, last lines, and pertinent intervening lines. Use vertical ellipses points to indicate any missing lines that users do not need to see.

- Examples can be included within a procedure (after a step) or at the end of a procedure.

---

**Note:** Be careful when providing examples of code or screenshots. Ensure that each name that you use for a URL, IP address, or network domain can be made public, or use the examples in this guide. See [Section 3.5.6](#).

---

## 9.3 Writing Procedures

A procedure is usually an ordered set of steps. However, a procedure can include only one step. A procedure can include prerequisites. A procedure can also be preceded by introductory text or by cross-references to overview or supplementary information.

A procedure can also be followed by one or more examples and by pointers to the next procedure or the next topic that must be addressed.

To write effective procedures, follow these guidelines:

- Introduce procedures with an infinitive phrase.
- Write procedures that are easy to follow.
- Place procedures appropriately.
- Use one method to describe a single procedure.

### 9.3.1 Introduce Procedures with an Infinitive Phrase

Begin the steps of the procedure with an infinitive phrase, such as: "To create a message:"

### 9.3.2 Write Procedures That Are Easy to Follow

To help users understand and follow procedural content, use these guidelines:

- Ensure that you establish the entire context in which the procedure is done.

You cannot assume that users have read surrounding paragraphs or procedures preceding the current procedure. Do not assume that users have already opened the screen or the file that previous sections or procedures discuss.

- Try to write no more than 10 steps for each procedure.

If a procedure is a long, single series of steps, then the procedure might be too complex. If a procedure is too long (more than 10 steps), then consider dividing the task into two or more smaller procedures.

Do not break up a long procedure if you cannot logically divide the steps. Therefore, do not separate a single procedure into smaller, less comprehensible procedures just to meet the recommended number of steps.

- Do not number single-step procedures. Place the single-step procedure in a separate paragraph.

- Include any prerequisite information.
- Provide explanatory text and visual cues.
- Include all required steps.

Users do not like to search through pages or go to another document to find required steps. In such cases, duplicate the required steps.

However, if procedural content is common to many procedures in a task, then include the procedure at the start of the task. Then, in subsequent procedures, cross-reference the common procedure.

- Do not provide a detailed description of each window, menu, or field in an interface.

Most users explore an interface with a specific task in mind. Therefore, in procedures, describe only the parts of the interface that are necessary to complete the task.

- Do not use graphics in place of procedural information.

A graphic that provides an overview of the areas in a window can help orient users. However, do not provide graphics of the interface in place of step-by-step procedures. In addition, do not provide procedural information in the callouts of a graphic.

- Do not repeat overview information or information that is not related to the task.

Place overview information in a section or in paragraphs before the procedure. Cross-reference any related, detailed supplementary information that supports the procedure.

- Add one or more examples to procedures if doing so can help the user. For more information about examples in procedures, see [Section 9.2.3](#).

### 9.3.3 Place Procedures Appropriately

To place procedures appropriately and consistently in a document, follow these guidelines:

- Put one or more procedures inside a section.

Place any introductory text or overview information that relates to one or more procedures under a section heading.

If introductory text is appropriate, then use one of these constructions:

- Full paragraph
- Complete sentence that ends with a period
- Complete sentence that ends with a colon

- Try to put procedures under a first-level section or a second-level section heading.

Place related procedures at the same level.

- Do not nest a procedure within another procedure.

### 9.3.4 Use One Method to Describe a Procedure

If users can perform a procedure in more than one way, then show only one method in a procedure. For example, do not mix steps that use a command-line interface with steps that use a graphical user interface in the same procedure.

Choose one method of presentation, command-line interface or graphical user interface, that best suits the needs of your users and the organization of the document.

An alternative is to present each method separately. Some of the more common ways to present each method separately are as follows:

- Put command-line procedures in one chapter and graphical user interface procedures in another chapter. Ensure that each chapter title identifies the specific method.
- Put related command-line procedures in one section and related graphical user interface procedures in another section.
- Primarily use the graphical user interface method, showing pertinent screenshots of graphical user interface windows within the procedure.
  - Use screenshots as a supplement to steps, not as a substitute for steps.
  - Do not overuse screenshots. Graphical user interfaces change frequently, and the screenshots may become a maintenance problem.
  - Provide visual cues only as necessary, and explain what happens after each step, if appropriate.
- If the entire procedure could be performed using the command line, then include a command-line example that shows the same actions at the end of the procedure.
- If only one step has a command associated with it, then consider adding a note or tip following the step that shows the command, for example:
  1. Click in the text where you want the symbol to appear.
  2. From the **Items** menu, select **Symbol**.

**Tip:** Alternatively, you can display this window by entering `show symbols` at the command line.

3. Double-click the symbol name.

If you provide both graphical user interface and command-line information, then provide users with enough information to choose one method over the other method.

## 9.4 Writing Steps

When writing steps, determine what a user must do first, next, and last. To write concise steps, follow these guidelines:

- Order and number the steps.
- Make each step short.
- Write each step as a complete sentence and in the imperative mood.
- Orient the user first with either the location (in a graphical user interface) or provide the goal of the task, and then tell the user what action to take.
- Write meaningful steps.
- Use branching of steps appropriately. For more information about branching, see [Section 9.4.5](#).

### 9.4.1 Order and Number the Steps

When ordering and numbering steps, follow these guidelines:

- Present information in a logical order. For example, ensure that a step that requires certain information appears after the user acquires that information.

Correct	Incorrect
1. Determine the new host's name. #sge-root/utilbin/SARCH/gethostname 2. In the sge-root/default/common/act_qmaster file, replace the current host name with the new host's name.	1. In the sge-root/default/common/act_qmaster file, replace the current host name with the new host's name.  This name should be the same as the name returned by the gethostname utility. To get the name, enter the following: #sge-root/utilbin/SARCH/gethostname

- Do not use a number if a procedure has only one step. Make that step a paragraph.
- If a procedure includes two or more steps, then use numerals to number the steps. Use letters for sequential substeps.
- Indicate optional steps with the word *Optional* within parentheses at the beginning of the step. Do not identify a step as optional if any user needs to complete the step for the procedure to be successful.  
3. (Optional) Reboot the system.
- If users must perform different actions depending on the outcome of a step, then use a bulleted list to show the alternatives. Use letters for substeps.

## 9.4.2 Make Each Step Short

A user can more easily follow a procedure when each step is short and explains one action. To help a user understand what to do in each step, follow these guidelines:

- Try to use no more than 20 words to write each step.
- Place any explanatory text in a separate paragraph under the step text, and keep the explanatory text as short as possible.
- Do not bury steps in a paragraph.
- Write about only one action in each step.

Exceptions to this guideline include the following:

- You conclude a step with *and click OK* because that action is a necessary component of the step.
- You begin a step with a common instruction such as *Log in as* or *Log in to* followed by another short instruction.

However, this combination of steps depends on the audience and the subject matter. Novice users might need more instruction than experienced users.

Novice User	Experienced User
1. Become a superuser. \$ su Password: 2. Reboot the system. # reboot	1. Become a superuser and reboot the system.

- Ensure that you include steps for all actions that the user must perform.

### 9.4.3 Write Each Step as a Complete Sentence in the Imperative Mood

Verbs do most of the work in instructions. When writing steps, follow these guidelines:

- Write each step as a complete, correctly punctuated sentence.
- Present the step as a statement of action rather than a question.
- Ensure that each step contains an active verb in the imperative mood. Reserve participles and gerunds for lists.

Put the verb at the start of the step unless you are explaining why, how, or where an action takes place. You might clarify a step to do one of the following:

- Qualify the verb.

Gently lift the I/O board up and out of the unit.

- Provide information to orient readers.

In the **Add Attachments** window, click **Add File**.

- State a condition.

If the card's I/O address conflicts with another device, then change the I/O address according to the manufacturer's instructions.

- Show the desired outcome or reason for the action.

To secure the board to the unit, tighten both screws.

- Stress the importance or consequence of an action.

To shut down the system, enter `shutdown`.

- Do not use command names as verbs.

Correct	Incorrect
1. To change to the new directory, enter <code>cd directory-name</code> .	1. <code>cd</code> to the new directory.

### 9.4.4 Write Meaningful Steps

To write complete steps that are effective, do the following:

- For GUI procedures, clearly state what users must do to interact with the interface.  
State what data the user must enter in a text field, which menu option the user must select, which button the user must click, and so on.
- For command-line procedures, make the task, not the command, the focus of the step. Follow the step immediately with the command syntax, if applicable.
- For command-line procedures, follow the step and the command line with a description of the command options and variables that directly relate to the step.

---

**Note:** If the command line contains only one or two self-explanatory variables and no command options, then you can choose not to define the variables. For example, if `filename` and `username` are the only variables, then you can choose not to define these variables.

---

- Explain to users why they are to skip a step or jump to a step.
  1. Determine whether you want the partition table to be the current table.
    - If you want to change the displayed partition table, then enter `n` and go to Step 4.
    - If you want to use the current partition table, then enter `y` when prompted.
- When providing an instruction in a step, ensure that you provide all the information that the user must have to complete the step. For example, if you instruct the user to stop a server or to edit a file, then provide the information about how to do so.

### 9.4.5 Use Branching of Steps Appropriately

Use branching if the action to take in a particular step in a procedure differs depending on the user's situation or intended outcome. Follow these guidelines to determine whether a step requires the use of branching:

- Use branching if the procedure is the same for many cases and differs only at one or two steps.
  1. Format the diskette:
    - To format the diskette for a UFS file system, enter `fdformat` and press Return.
    - To format the diskette for a Windows file system, enter `fdformat - w` and press Enter.
- Indicate the branching condition in the main step text, not by using the word *or*.

Correct	Incorrect
To restart NFS, enter one of the following commands:	1. Restart NFS.
■ <code>/etc/init.c/nfs/restart</code>	<code># /etc/init.d/nfs/restart</code>
■ <code>/etc/rc3.d/s60nfs restart</code>	OR
	<code>#/etc/rc3.d/s60nfs restart</code>

- If the branching condition applies to most or all of the procedure, then use two different procedures.
 

For example, if the procedure has several steps that provide alternatives for HTTP and FTP protocols, then create two procedures. Write one procedure for HTTP and one for FTP.
- If a particular condition requires a substitution in most of the steps in the procedure, then provide that information in a note.
 

For example, tell users to use the default directory if their default directory is different from the directory provided in the steps.
- If a step is optional, then do not use branching.



Correct	Incorrect
1. (Optional) To make this printer the default printer, select <b>Default</b> .	Determine whether you want to make this printer the default printer. <ul style="list-style-type: none"> <li>■ If no, then go to Step 2.</li> <li>■ If yes, then select <b>Default</b>.</li> </ul>
<ul style="list-style-type: none"> <li>■ If a user must know certain information to determine which branch to follow, then include the process by which the user can find out that information.</li> </ul>	
Correct	Incorrect
1. Use a text editor to ensure that the .html file is complete. <ul style="list-style-type: none"> <li>■ If the file is complete, then post the file on the internal website.</li> <li>■ If the file is not complete, then see Appendix C.</li> </ul>	1. Ensure that the .html file is complete. <ul style="list-style-type: none"> <li>■ If the file is complete, then post the file on the internal website.</li> <li>■ If the file is not complete, then see Appendix C.</li> </ul>

## 9.5 Checking for Structural Problems

This section describes some signs of a possible need for restructuring.

### 9.5.1 Duplicate Series of Steps

If two or more procedures begin with the same series of steps, then consider creating a separate procedure with the shared steps and then cross-referencing to that procedure in the related procedures.

For example, suppose you have several procedures that are accomplished through a web page deep in the application's hierarchy. To describe how to get to the Modify Objects page requires four steps. You can create a separate procedure. Then, at the beginning of each modification task, Step 1 could be *If you are not already on the Modify Objects page, then see "Accessing the Modifying Object Page."*

### 9.5.2 Nearly Identical Procedures

Look for two or more procedures that are alike except for one or two steps that require a different value or choice. Consider combining the procedures into one procedure. Provide information in the steps that require choices.

For example, a word processing application might use the same basic procedure to create generated lists such as tables of contents, lists of tables, and lists of figures.

Rather than having a separate procedure for each type of generated list, you can provide one procedure that describes how to create generated lists. Then, in the relevant steps, you can provide the specific file names related to the type of list that the user is creating.

### 9.5.3 Many Nested Substeps

Multiple levels of substeps within several steps in a procedure probably indicate that the procedure should be divided into separate procedures.

### 9.5.4 Procedures with More Than 10 Steps

Long procedures are difficult to follow. Look for a logical place to divide the procedure. You might want to describe the overall procedure, followed by a task map or numbered list that describes the related procedures.

### 9.5.5 Several Single-Step Procedures

The presence of many single-step procedures might indicate some different structural problems:

- If many of the procedures describe the same basic action, then you might be able to collapse the procedures.  
  
For example, suppose you have separate single-step procedures for opening different applications from a front panel. You might want to provide one single-step procedure with the heading *Opening an Application from the Front Panel*. If necessary, then provide a cross-reference from each application.
- If some procedures are related logically, then you might be able to combine the procedures.  
  
For example, suppose you must add one line to a system file to set a printer resource and another line to set a scanner resource. You can provide a procedure with the heading *Adding Peripherals to the .Nresource File*. The procedure contains the steps that are common to the procedures that are being combined. Then one step includes the different text lines for each peripheral.
- Examine each single-step procedure to ensure that all required steps are provided.  
  
For example, you might be assuming that the user is at a particular place in the GUI or has already logged in to the system.

### 9.5.6 Repeated Steps Indicated at the End of Repeated Actions

Instructions such as *Repeat Steps 2 through 4* that are placed at the end of a set of steps can indicate a structural problem. Instead, indicate the repetition at the start of the steps to be repeated. Sometimes, this strategy results in nested substeps, but it also shortens the overall number of steps and gives the user sufficient context in the form of an advanced warning.

Correct	Incorrect
1. On each system perform the following steps:  a. Determine which data links are configured on a host. # dladm show-link  b. Identify which VID to associate with each data link on the system.  c. Create PPAs for each interface to be configured with a VLAN.	1. Determine which data links are configured on a host. # dladm show-link  2. Identify which VID to associate with each data link on the system.  3. Create PPAs for each interface to be configured with a VLAN.  4. Repeat Steps 1 through 3 on each system.