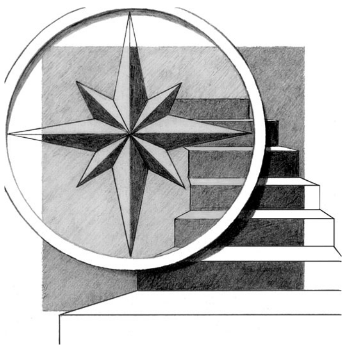


Chapter 2 Task orientation

Don't tell me how it works, tell me how to use it. —A customer

Task-oriented writing is writing in terms of how the user does the task. You rarely help your users when you tell them only how a product works or how it is structured internally. Your users have a job to do, so they need practical information—how-to information.

You need to understand the tasks that you're writing about from your users' perspective. Do a task analysis to determine which tasks are most important to each group of users, which tasks are most frequent, and which tasks are most difficult. Make a list of the high-level tasks that users will do with your product.

You can divide high-level tasks, such as getting started with the product, into groups of lower-level tasks, such as installing the product and setting up the product. Each of these tasks might also be divided into still lower-level tasks until you have groups of manageable tasks that make sense to the user. For example, opening a bank account is a discrete task that makes sense for a user to do. However, the action of typing an address might be a step of a task, but it is not a task on its own.

Task topics, whether high-level or low-level, are the most important types of topics for users because tasks help users do their jobs. Users are frustrated if they cannot complete a task. Task-oriented topics get the user back "on task." Like a compass on a journey, tasks provide direction.

To make information task oriented, follow these guidelines:

- ❑ **Write for the intended audience.**
- ❑ **Present information from the user's point of view.**
- ❑ **Indicate a practical reason for information.**
- ❑ **Focus on real tasks, not product functions.**
- ❑ **Use headings that reveal the tasks.**
- ❑ **Divide tasks into discrete subtasks.**
- ❑ **Provide clear, step-by-step instructions.**

Write for the intended audience

Before you start writing, be sure that you have a clear understanding of your audience. For example, if you are writing for managers, you might include only high-level tasks, such as evaluating and planning, or a high-level view of other tasks. Similarly, if you are writing for end users, avoid system administrator tasks.

Be sure that the information that you include in your topics is of interest to your audience. For example, your product might have a powerful new help system, but a description of the help system features is of little interest to the person who is installing the product.

The following passage shows a simple task that is explained in detail. However, the audience consists of users who want to use an advanced feature and therefore do not need help performing simple tasks. This information will frustrate all but the most patient advanced user.

Original

To customize your settings:

1. Go to the file tree.
2. Click the **INFODIR** folder.
3. Right-click the SETTINGS.DEF file and select **Edit** from the menu.
4. Change the settings that you want in the file.
5. Click **File —> Save** to save the file.
6. Click **File —> Close** to close the editor.

Revision

To customize your settings, edit the INFODIR/SETTINGS.DEF file.

In the revision, the task is handled much more simply. The revision quickly provides the users with the information that they need, because the writer understands the skill level of the audience.

For more information about how much detail to provide based on the type of user, see the completeness guideline "Cover each topic in just as much detail as users need" on page 79.

Present information from the user's point of view

Writing from the user's point of view brings the user into the "story," so that users can easily imagine doing the task that you describe. Such writing has these characteristics:

- ❑ It is predominantly directed at "you" (second person).
- ❑ It uses the active voice, with verbs that denote actions that the user does as opposed to actions that the product does.
- ❑ It gives a reason for the actions.

The following passage is written from a remote, impersonal point of view:

Original

The system should not be shut down during processing. If such a shutdown occurs, the system should be restarted with the START RECOVER command.

Revision

If you shut down the system during processing, you might lose data. Use the START RECOVER command to restart the system and recover any data from the log.

The original passage is passive and indefinite about who does the action and why. In the revision, the information is presented to make the user an active participant. The phrase "you might lose data" expresses the reason for the action in terms that users can relate to personally—they don't *want* to lose data.

The following passage leaves the users out of the story altogether:

Original

Subsequent installation of the HIGS feature allows InfoProduct to run unattended.

Revision

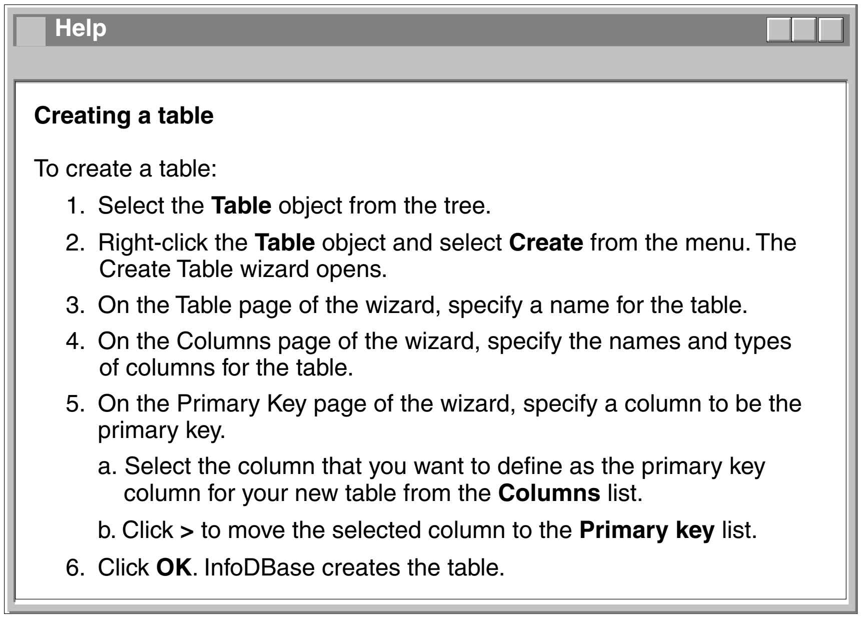
If you want to run InfoProduct unattended, you must install the HIGS feature. You can install the HIGS feature after you install InfoProduct.

Users don't know how the information in the original passage pertains to them. The revision adds the users to the story and explains what they need to *do* to use the feature.

When you provide task help, you have the advantage of knowing where the users are in the product interface and, therefore, what part of the task they need help with. This advantage allows you to write a help topic that's specific to the user's position.

For the following help topic, assume that a user accesses online help from the Primary Key page (which is the third and final page) of the Create Table wizard.

Original



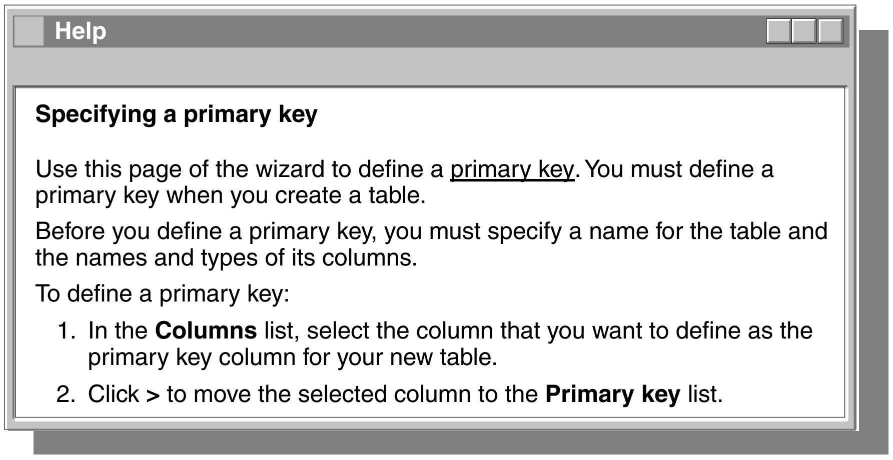
Creating a table

To create a table:

1. Select the **Table** object from the tree.
2. Right-click the **Table** object and select **Create** from the menu. The Create Table wizard opens.
3. On the Table page of the wizard, specify a name for the table.
4. On the Columns page of the wizard, specify the names and types of columns for the table.
5. On the Primary Key page of the wizard, specify a column to be the primary key.
   a. Select the column that you want to define as the primary key column for your new table from the **Columns** list.
   b. Click **>** to move the selected column to the **Primary key** list.
6. Click **OK**. InfoDBase creates the table.

The user has already completed most of the steps of the "Creating a table" task and needs only information about how to specify a primary key, which is a subtask of creating a table. The original help topic wastes the user's time because it provides steps that the user has already completed.

Revision



Specifying a primary key

Use this page of the wizard to define a primary key. You must define a primary key when you create a table.

Before you define a primary key, you must specify a name for the table and the names and types of its columns.

To define a primary key:

1. In the **Columns** list, select the column that you want to define as the primary key column for your new table.
2. Click **>** to move the selected column to the **Primary key** list.

The revised help topic presents the task from the user's probable position. The revised topic also provides a link to a definition of a primary key for users who need to understand that concept before proceeding with the task.

When you understand the user's point of view, you can write task-oriented information that helps users do their tasks.

Indicate a practical reason for information

Giving users the information that they need is only part of task-oriented writing. Users need a practical reason for the information. They need to understand *why* you are giving it to them—how it is relevant to their task. The goal that the information serves must be apparent. A task is not just an activity, but an activity that is directed to a particular end.

To ensure that the information that you provide is relevant to the task, follow these guidelines:

- ❑ Relate details to a task where appropriate.
- ❑ Provide only a necessary amount of conceptual information in task topics.

Relate details to a task where appropriate

In a task topic, users should never wonder, "But why are you telling me this?" For example, to state that the records in a file have a certain size might leave users wondering, "So what?" However, if you tell them that they must build a library to hold the file and that the record size affects the way that they do this, they can understand why you are telling them about record size.

In a task topic, facts can puzzle users if you don't indicate what significance the facts have, as shown in the following passage:

Original

If the NORES option is used, the routines are link-edited as part of the load module. If the RES option is used, the routines are loaded separately.

Revision

Use the NORES option when you have sufficient space for routines to be link-edited as part of your load module. Use the RES option to save space by loading the routines only when you need them.

The original passage explains what the options do, but it does not relate that information to the user's task of deciding which option to use. In the revision, the facts are restated so that the user understands when to use which option.

At first glance, the following sentence appears to be only descriptive and to have no practical application:

Original

> The BW_Message mapping table in the Data_LM directory can contain warning messages that are issued by InfoProduct when you create a request.

Revision

> After you create a request, check the BW_Message mapping table to see if InfoProduct issued any warning messages. The BW_Message mapping table is in the Data_LM directory.

Users might glance at the original sentence and move on because it doesn't seem relevant. The revised sentence relates the information to the task of creating a request, so that users understand why the information is significant.

Provide only a necessary amount of conceptual information in task topics

Task topics should focus on providing only the information that supports the task. Task topics can contain:

- ❑ Rationale for doing the task
- ❑ Requirements for doing the task (such as prerequisite tasks or software)
- ❑ Steps of the task
- ❑ Tips and examples to support the steps
- ❑ Information about follow-on (postrequisite) tasks

Do not mix significant amounts of conceptual information with task information. Users want to learn to do things without needing to learn all the concepts that are associated with a product. In task topics, provide steps as early as possible, and explain minor concepts briefly, if at all.

Separate the explanations of major concepts from the task topic by creating a topic for each major concept. Provide links to those concept topics from your task topic. That way, users who already understand the concepts can do the task, and users who need to learn about the concepts can go to the concept topics.

This guideline deals with overcompleteness and is similar to the completeness guideline "Include only necessary information" on page 84.

The following topic explains the concepts of a project and a suite before presenting the task:

Original

Creating test cases

Projects are collections of files that are related to a test case. When you create a test case, you must also create a project. You can, however, create a project before you create a test case. Each project can contain multiple test case files but only one main test case. You create projects in suites. Each suite can contain multiple projects. If you create one suite for each function that you test, the projects in a suite can contain the test cases that you create to test the function.

To create a suite:

1. Right-click a suite object and select **New** from the menu.
2. Specify details in the New Suite window.

To create a project:

1. Right-click a suite object and select **New Project** from the menu.
2. Specify details in the New Project window.

To create a test case:

1. Right-click a project and select **New Test Case** from the menu.
2. Specify details in the New Test Case window.
3. Click **Record** and work with the product that you want to test.
4. Click **Save** after you finish recording your actions.

The original topic starts with conceptual information and then tells the user how to create the needed elements. Users need to read through a big paragraph of information before they can start the task.

Revision

Creating test cases

You can create one or more test cases for each function that you want to test.

Before you begin:

1. Create a suite for the function that you are testing.
2. Create a project within the suite to hold the test cases and files.

To create a test case:

1. Right-click a project and select **New Test Case** from the menu.
2. Specify details in the New Test Case window.
3. Click **Record** and work with the product that you want to test.
4. Click **Save** after you finish recording your actions.

Related topics
Creating suites
Creating projects
Suites
Projects
Test cases

The revised topic does not explain all of the relationships of the elements. These elements are explained in detail in concept topics. Links are provided from the bottom of the task topic to these concept topics. In addition, the revised topic uses a prerequisite section that is labeled "Before you begin" to show that two of the elements must be created before you begin this task.

This separation of task topics and concept topics allows the writer to share the same concept topics across multiple tasks. For example, the task topic for creating a project can also provide a link to the "Projects" concept topic.

The revised task does not provide the substeps of the prerequisite tasks. Those steps are provided in separate tasks that are linked to from this task. Each of the three tasks is handled in a separate topic as explained in the guideline "Divide tasks into discrete subtasks" on page 33 of this chapter.

Focus on real tasks, not product functions

A *real task* is a task that users want to perform, regardless of whether they are using your product to do it. Tasks that are imposed by the product are *artificial tasks*. It is all too easy in technical writing to lose sight of the real tasks and get caught up in the tasks that are dictated by the product. When you live and breathe a product for months, you might forget that the user's tasks and the product's tasks are not necessarily the same.

Examples of real and artificial tasks are:

- ❑ Users want to edit a table, but the writer introduces this task as "using the table editor" instead of "editing a table."
- ❑ Users want to count the records in a file, but the writer introduces the task as "using the CNTREC utility" instead of "counting records with the CNTREC utility."

Sometimes the design of a product forces you to write about artificial tasks. For example, users want to create a graphic, but they must first set up a container to hold sets of graphics. In such cases, you can focus on the task instead of the design by writing about "organizing your graphics with containers" rather than "creating containers."

In the following lead-in to the steps, the writer makes the common mistake of describing the task in product-specific terms:

Original

To use the InfoInstaller utility:

1. Open the InfoInstaller window by typing `infoinst` at the command line.
2. Complete the InfoInstaller window by specifying the installation parameters.
3. Click **OK**. InfoInstaller installs InfoProduct.

The original introduction assumes that users understand the task in terms of the tool that they need to use to do the task. Although some users might know what InfoInstaller is, all users know what installation is.

Revision

To install InfoProduct:

1. Type `infoinst` at the command line. The InfoInstaller window opens.
2. Specify the installation parameters in the window.
3. Click **OK**. InfoProduct is installed.

The revised introduction and steps separate the task from the tool so that users can relate to the real task of installing the product instead of to the tool that they use to do so.

The following introduction shows a topic that explains how to use the product rather than how to perform real tasks:

Original

This topic explains how to use the following menu choices under **File**:

Open	Opens an existing file.
New	Creates a file.
Save as	Saves to a new file with a different name.

The original text assumes that the user is examining the interface and wondering what each menu item does. This type of information is appropriate for *contextual help* (help that is relevant to where a user is in a product, such as help for the selected control), but not for a task topic. Users want information about how to do real tasks, not a list of the buttons and fields in the product interface.

Revision

This topic explains how to work with a document. You can do the following tasks:

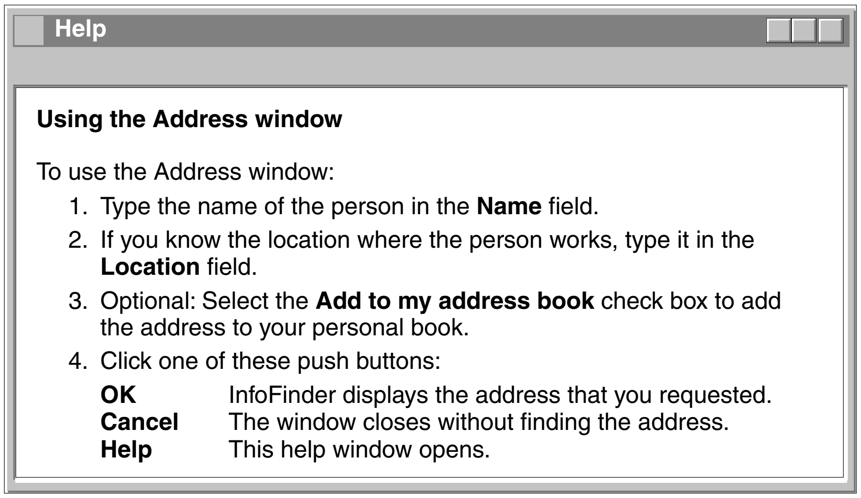
- ❑ Create a document
- ❑ Open an existing document
- ❑ Rename a document

The revision shows information that is presented in terms of real tasks.

By presenting the information from the perspective of tasks that users recognize and want to perform, you make the information more relevant to users.

The following topic presents the task in terms of the window that is used to do the task:

Original



Help

Using the Address window

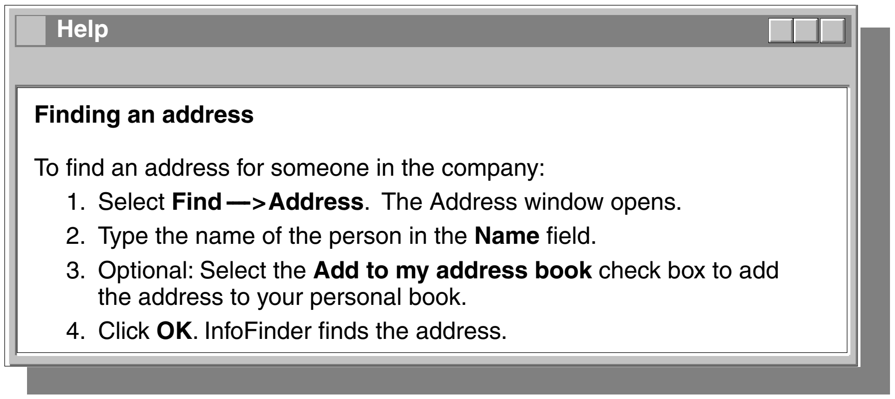
To use the Address window:

1. Type the name of the person in the **Name** field.
2. If you know the location where the person works, type it in the **Location** field.
3. Optional: Select the **Add to my address book** check box to add the address to your personal book.
4. Click one of these push buttons:

OK	InfoFinder displays the address that you requested.
Cancel	The window closes without finding the address.
Help	This help window opens.

The original help topic tells the user how to use the window; it explains how to complete the fields in the window, but it never explains what real task the user is doing. The original help topic also contains extraneous information about the window that isn't pertinent to the real task.

Revision



Help

Finding an address

To find an address for someone in the company:

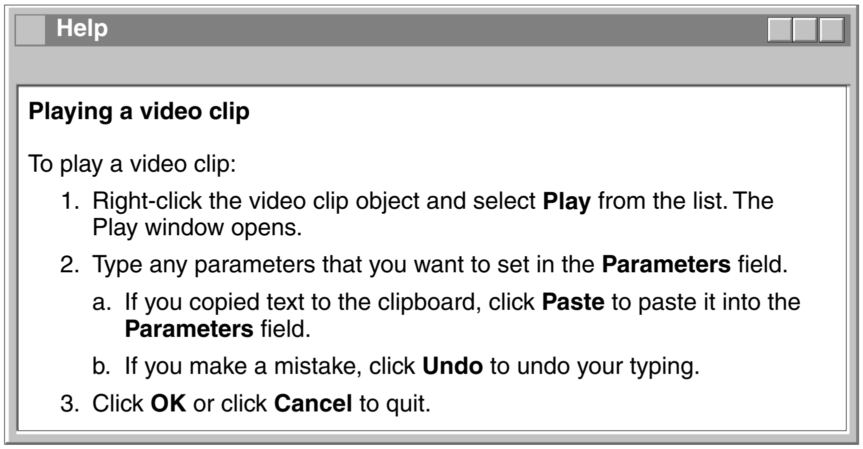
1. Select **Find —>Address**. The Address window opens.
2. Type the name of the person in the **Name** field.
3. Optional: Select the **Add to my address book** check box to add the address to your personal book.
4. Click **OK**. InfoFinder finds the address.

The revised help topic presents the information in terms of the task that the user wants to do and the steps that are involved in the task.

Keep task steps focused on the real task. Avoid littering steps with feature "advertisements" unless the features are especially helpful to the user as part of the task. Apply the completeness guideline "Cover each topic in just as much detail as users need" on page 79.

The following help topic contains unnecessary information about features that is mixed in with the task:

Original



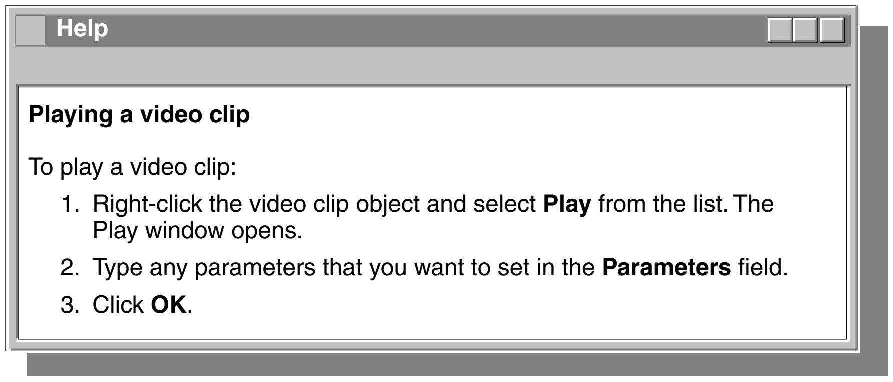
Help

Playing a video clip

To play a video clip:

1. Right-click the video clip object and select **Play** from the list. The Play window opens.
2. Type any parameters that you want to set in the **Parameters** field.
   a. If you copied text to the clipboard, click **Paste** to paste it into the **Parameters** field.
   b. If you make a mistake, click **Undo** to undo your typing.
3. Click **OK** or click **Cancel** to quit.

Revision



Help

Playing a video clip

To play a video clip:

1. Right-click the video clip object and select **Play** from the list. The Play window opens.
2. Type any parameters that you want to set in the **Parameters** field.
3. Click **OK**.

Steps 2a and 2b in the original help topic are *not* part of the task. Clicking **Cancel** is also not part of the task. These are only features that users can use. If you document every feature like this, your tasks will become unwieldy and lose their focus. The revision shows the real steps of the task.

Always focus on real tasks, not artificial tasks, when you write task topics, headings, and steps.

Use headings that reveal the tasks

The first place to tell users why they are being given information is in the heading. Users should get an accurate idea about the content of the topic from the heading. Each heading should reveal that the topic contains information about a task, and what that task is.

A static heading like "Authorizations" might be fine for a concept or reference topic, but for a task topic, the heading should reveal the task that users are being told how to do. A more helpful heading might be "Authorizing users" or "Setting authorizations for a user ID."

Be careful not to mislead users by using *pseudo-task* headings. Pseudo-task headings start with vague verbs, such as "understanding" and "learning" (and sometimes "using"). Pseudo-task headings mislead users in the following ways:

- Pseudo-task headings make a topic appear to be task-oriented when it is actually full of conceptual or reference information. For example, a topic called "Understanding the file system" probably does not contain steps for understanding the file system, but instead contains reference information about the file system. A more appropriate heading might simply be "The file system."
- Pseudo-task headings substitute artificial (product-imposed) task headings in place of real task headings. For example, a heading like "Using the SpellMaster tool" is hiding the real task, and should probably be called "Checking the spelling in a document" instead. The guideline "Focus on real tasks, not product functions" on page 27 of this chapter provides more discussion and examples of real and artificial tasks.

Headings are elements that help users find information. Good headings produce a good table of contents, from which users can predict what they will find in each topic: task, conceptual, or reference information. Therefore, this guideline is related to retrievability as well as task orientation.

The following headings are misleading and unhelpful:

Original

Register usage
Administering authorization
The Dial-up function
Session initialization
Using the Define Font Window
Understanding hardware requirements

Some of the original headings hide the fact that the topic contains task information; others define the tasks in terms of the product; and the last heading misrepresents a reference topic as a task.

Revision

Linking with registers
Authorizing access to data
Dialing up the computer
Initializing a session
Defining a font
Hardware requirements

The revised headings clearly reveal the type of information that each topic contains; they use verbs that express user actions, and they indicate the goal of the action.

The revised headings use gerunds for task-oriented headings. Gerunds are not the only alternative for task-oriented headings. Your style guideline might call for headings like “How to define a font” or “Steps for defining a font” instead.

The following excerpt from a list of topics doesn’t give the user any idea of what task the information supports:

Original

- Working with InfoDataMagic
  - Using collectors
    - Starting collectors
    - Configuring collectors
    - Stopping collectors
  - Using repositories

Revision

- Managing copied data
  - Collecting data
    - Starting a data collection
    - Configuring collection properties
    - Stopping a data collection
  - Storing data

In the original list of topics, the headings could apply to a variety of tasks. Users who are not familiar with the terminology for the product have no idea what tasks the topics explain and could easily miss these sections in a list of topics. The revised list of topics shows task-oriented headings that should make sense to all users.

Divide tasks into discrete subtasks

After you identify your main tasks, you need to divide them into discrete subtasks so that you can provide usable step-level information. If you tried to provide step-level information for the main task of writing a news story, you would have thousands of steps. Instead, you divide the task of writing a news story into its main subtasks, for example, researching the material, drafting an outline, writing a first draft, and revising the draft. Repeat the process of dividing tasks until you have groups of tasks for which you can provide step-level information.

Keep in mind that the number of steps in each task and subtask should be nine or fewer, as discussed in the clarity guideline "Keep elements short" on page 124. If you find that a task takes more than nine steps to do, determine whether the task can be divided again or whether some steps can be nested under others. For more information about nesting steps, see the guideline "Group steps for usability" on page 38 of this chapter.

Before you write task information, you need to know how the task is related to other tasks. You need to understand the task sequences, levels of tasks, and interdependencies. Is the task really a discrete task? Does it have subordinate tasks? Is it a subtask of a larger task? Is it a subtask of more than one task?

Although this guideline applies to task orientation, it does overlap with the following organization guidelines: "Organize information into discrete topics by type" on page 218 and "Organize tasks by order of use" on page 222.

The following task overview is organized according to the order that the user should do the subtasks. You can easily replace this overview with a high-level task that lists appropriate subtasks.

Original

To migrate to version 8, you need to read these topics:

❑ Hardware requirements
❑ Software requirements
❑ Applying the latest updates
❑ Stopping InfoProduct processes
❑ Backing up your infoproduct.inf file
❑ Running the migration utility for Windows
❑ Running the migration utility for UNIX
❑ Verifying migration
❑ Setting up a new profile

Revision

To migrate to version 8:

1. Ensure that you have the correct hardware and software:
   ❑ Hardware requirements
   ❑ Software requirements
2. Apply the latest updates.
3. Stop all InfoProduct processes.
4. Back up your infoproduct.inf file.
5. Run the migration utility.
   ❑ On Windows
   ❑ On UNIX
6. Verify the migration.
7. Set up a new profile.

After you finish migrating InfoProduct, you can start the processes again.

The revision shows the relationship of the subtasks to each other and to the higher-level task. The revision also mentions what to do after completing the task.

The following list of topics shows a hierarchy of tasks in which the task of verifying the installation is combined with the task of configuring parameters:

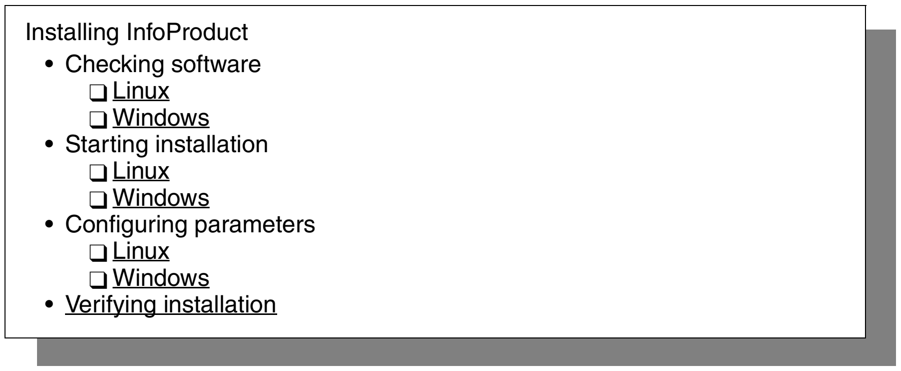
Original

Installing InfoProduct

- Checking software
  ❑ Linux
  ❑ Windows
- Starting installation
  ❑ Linux
  ❑ Windows
- Configuring parameters and verifying installation
  ❑ Linux
  ❑ Windows

In the original list, the task of verifying installation is repeated for both Linux and Windows operating systems. The task of verifying installation is not part of the task of configuring parameters. One user might verify the installation while another user configures the parameters. The two tasks should not be combined.

Revision



Installing InfoProduct

- Checking software
  - ❑ Linux
  - ❑ Windows
- Starting installation
  - ❑ Linux
  - ❑ Windows
- Configuring parameters
  - ❑ Linux
  - ❑ Windows
- Verifying installation

The revision shows verifying installation as the last, discrete task at the same level as the other main subtasks of installing the product. The revision allows for a logical separation of different tasks.

Consider the case in which a user might need to configure additional parameters at some time other than during installation. Seeing the steps for verifying installation with the steps for configuring parameters does not make sense to that user. In addition, if the steps for verifying installation are the same on Linux as those on Windows, you can write the information once instead of repeating those steps in the topic for Linux and the topic for Windows.

Be sure to separate your tasks into discrete subtasks such that each subtask contains only one distinct task, yet contains sufficient content to stand alone.

Provide clear, step-by-step instructions

Steps make up most tasks. Occasionally a task has only one step and can be described in a paragraph, but most tasks are performed as a series of ordered steps. Task orientation extends to the level of the lowest step. Any step that is not clearly written or not ordered correctly can cause your users to make mistakes and be unable to do a task.

When you write task information, you are usually in a position to notice usability problems in time to suggest product improvements. For example, you might identify cumbersome steps that can be streamlined or avoided, or steps where users are repeating actions that they've already performed (such as typing a long serial number in more than one window). If you have difficulty documenting a task, consider whether there might be a problem with the way that the product works. Keep the needs and interests of your users in mind as you write; there is no such thing as a product that is too usable. In fact, the more usable a product is, the less users must rely on low-level step information.

Take the time to organize your steps from your users' perspective. Know the answers to these questions: How do tasks relate to each other? When are steps subtasks? What constitutes a step? Where does a step start and end? Are some steps subordinate to others? Which steps are optional? Which steps are conditional?

The following guidelines can help you provide clear step-by-step instructions:

- ❑ Make each step a clear action for users to take.
- ❑ Group steps for usability.
- ❑ Clearly identify optional steps.
- ❑ Identify criteria at the beginning of conditional steps.

Make each step a clear action for users to take

Each step should correspond to an action (high level or low level) that the user performs. Tasks do not discuss how the user and product interact; tasks list only the actions that the users do to complete their task.

A step isn't complete unless it has an action for the user to do. One of the following steps has no user action:

Original

3. Click **OK**.
4. The installation begins.
5. After installation completes, restart your system.

Revision

3. Click **OK**. The installation begins.
4. After installation completes, restart your system.

In the original set of steps, step 4 describes a product action, not a user action. In the revision, step 4 is combined with step 3 because it is the result of step 3, not a separate step.

To ensure that each step gives clear direction to users, include an imperative verb (a verb that instructs the user to take an action) in the first sentence of every step. When you make style decisions for your product, you might pick a specific way to phrase step-level information. For example, you might choose to "place" your user before stating the user action, as in "In the first column of the table, type the date." Alternatively, you might choose to put the user action before the placement, as in "Type the date in the first column of the table." Both approaches include the imperative verb in the first sentence of the step.

Some decisions are trickier than others. Consider the following set of steps:

Original

3. Click **OK**.
4. The InfoUpdater should stop.
   - If it doesn't stop, repeat steps 2 and 3.
   - If it does stop, go to step 5.
5. Run the InfoVerify tool to check for viruses.

Revision

3. Click **OK**. InfoUpdater should stop.
4. If the InfoUpdater is not stopped, repeat steps 2 and 3.
5. Run the InfoVerify tool to check for viruses.

Step 4 of the original set of steps breaks the rule of including an imperative verb in the first sentence of every step. Step 4 is not a step. In the revision, step 4 includes an imperative verb and is easier for users to follow.

Group steps for usability

Group steps to help users relate to the task. If you instruct users to do one action or click after another, the task can become mind numbing for the user. If you can group minor steps together into a larger step, users can think of the steps in relation to the goal of completing the task.

For example, instead of interpreting the steps as "First I click here, then I fill out that field, then I click over there, then I choose a button here," users might be able to think of the steps in terms of "First I set my preferences, then I specify the server information." In this way, you not only help users relate to the steps that they are doing, you also streamline the steps and make each step easier for users to find.

The following steps are in the correct order, but they don't correspond to the way that the user thinks about the task:

Original

To add a setting to your profile:

1. Select the profile object that you want and right-click.
2. Select **Properties** from the menu.
3. In the Properties window, find the name and path of the profile file.
4. Close the Properties window.
5. Open your profile file in a text editor.
6. Add the setting to your profile file in the settings section.
7. Save the profile file.
8. Run the profile command with the -file *YourProfileName* option.

The original set of steps gives each step the same weight. It treats trivial steps, such as closing a window, the same way that it treats more significant steps, such as running the command. The original set of steps ignores the relationship of each step to its surrounding steps.

Revision

To add a setting to your profile:

1. Determine the name of the profile file that you want to add the setting to:

   a. Right-click the profile object that you want and select **Properties** from the menu.
   b. In the Properties window, find the name and path of the profile file.

2. Update the profile file with the new setting:

   a. Open your profile file in a text editor.
   b. Add the setting to your profile file in the settings section.
   c. Save the profile file.

3. Run the profile command with the -file *YourProfileName* option.

The revised set of steps shows the relationships of some of the steps to each other and shows how they make up the two higher-level steps: finding the name of the file and updating the file. The revised set of steps also downplays some of the trivial steps by merging them or omitting them. Where the original set of steps shows a linear progression of one action or click after another, the revised set of steps shows what each step accomplishes toward completing the whole task. Also, by combining the steps into higher-level steps, the revision minimizes the number of steps.

Unordered lists, or bulleted lists, provide another way to subordinate information or actions in steps. Be sure to use unordered lists only for tasks that are not sequential. You could use an unordered list to show, for example, more than one way to do a step.

The following set of steps uses unordered lists to try to get both a new user and a returning user through the first two windows needed for a task:

Original

To purchase tickets online:

1. Go to www.e-infoticket.com.
2. Select one of the choices.
   - If you are a new user, click **Register**.
   - If you registered before, click **I Have an Account**.
3. Complete the fields on the form.
   - If you are a new user, the form requires a credit card number.
   - If you are a returning user, specify your password.
4. Click **Select Tickets to Purchase**.
5. Choose the tickets that you want.
6. Click **Submit** and wait a few seconds for a confirmation message.
   - If a message tells you that the tickets are purchased, write down or print the confirmation number. You have finished.
   - If no message appears, repeat steps 5 and 6.

In the original set of steps, both the new user and existing user need to read both step 2 and step 3 to figure out what to do. The original steps are focused on the product, not the flow of the task.

Revision

To purchase tickets online:

1. Go to www.e-infoticket.com.
2. If you are a new user, set up an account:
   a. Click **Register**.
   b. Specify your credit card number and a name and password for your account.
   c. Click **Submit**. When the account is set up, a message will tell you to proceed.
3. Click **I Have an Account**.
4. Specify your name and password.
5. Click **Select Tickets to Purchase**.
6. Choose the tickets that you want.
7. Click **Submit** and wait a few seconds for a confirmation message.
   - If a message tells you that the tickets are purchased, write down or print the confirmation number. You have finished.
   - If no message appears, repeat steps 6 and 7.

In the revision, step 2 is used to prepare the new user for the remaining steps. Thus both types of users can follow one clear set of steps.

When you use sublists to group steps, be sure to use the right type of sublist for the situation. The following example shows a step divided into substeps:

Original

3. Copy the contents of the file system from the source disk to the target disk. The steps that you use depend on the location of the target disk:

   a. If the target disk is on the same computer as the source disk, run the copyfilesystem command.

   b. If the target disk will replace the source disk, follow these steps:

      1. Copy the contents of the source disk to tape.
      2. Replace the source disk with the target disk.
      3. Configure the target disk.
      4. Copy the tape contents to the target disk.

In the original step, ordered substeps are used to show two choices that are mutually exclusive. Because the choices are not meant to be performed in order, the substeps are misleading.

Revision

3. Copy the contents of the file system from the source disk to the target disk. The steps that you use depend on the location of the target disk:

   • If the target disk is on the same computer as the source disk, run the copyfilesystem command.
   • If the target disk will replace the source disk, follow these steps:

      a. Copy the contents of the source disk to tape.
      b. Replace the source disk with the target disk.
      c. Configure the target disk.
      d. Copy the tape contents to the target disk.

In the revision, the substeps are replaced with bulleted options. Users follow the instructions in one bullet or the other.

Clearly identify optional steps

Optional steps are steps that a user can skip and still complete the task successfully. As mentioned in the guideline "Focus on real tasks, not product functions" on page 27 of this chapter, try to keep your tasks free of feature clutter by eliminating steps that are superfluous to the task. However, in cases where optional steps support the task, include them in the task, but identify them as optional. For example: “2. Optional: Define a profile for your startup parameters.”

The following list of steps identifies step 1 as optional, but not step 2:

Original

1. Optional: Click **Timeout Settings** to specify timing settings for the profile. The Timeout Settings window opens.
2. In the Timeout Settings window, specify the number of seconds before the system is to restart.

The original set of steps is confusing. Users who choose to skip step 1 are unable to do step 2. Because step 2 can be done only if the user follows step 1, step 2 must also be optional.

First revision

1. Optional: Click **Timeout Settings** to specify timing settings for the profile. The Timeout Settings window opens.
2. Optional: In the Timeout Settings window, specify the number of seconds before the system is to restart.

The first revision shows both steps as optional. However, the first revision is still not logical because users who choose not to perform step 1 cannot perform step 2. So step 2 is not optional by itself.

Second revision

1. Optional: Specify timing settings for the profile:

   a. Click **Timeout Settings** to open the Timeout Settings window.
   b. Specify the number of seconds before the system is to restart.

The second revision shows step 1 as an optional step that consists of two substeps. If users choose to follow step 1, they must perform both steps 1a and 1b, which is the only combination that makes sense.

Take care to clearly identify optional steps. Use “Optional” for optional steps. However, do not follow the word “Optional” with the phrase “If you want to” or “You can” because these phrases are redundant with the word “Optional.”

Identify criteria at the beginning of conditional steps

Conditional steps are those that users follow only if certain criteria apply. Conditional steps generally begin with the word "If," as in, "If you run test cases in batch mode, complete the fields on the Batch page." Users who meet the criteria for the step must follow the step. Always start conditional steps with the condition. That way, users who do not meet the criteria can skip the step after reading the condition.

Although the following steps are conditional, users might find themselves halfway through the steps before they realize that they don't need to do them.

Original

1. Register your computer as a client if you are not yet registered on the LAN.
2. In the **Number** field, specify your 12-digit serial number if your software is not yet registered.
3. Run the InfoExec program to reconfigure your settings. (InfoExtended only)

Because users rarely read ahead when following steps, the original steps might cause some users to take links that do not apply to them, start typing serial numbers, or try to run programs that they don't need.

Revision

1. If you are not yet registered on the LAN, register your computer as a client.
2. If your software is not yet registered, in the **Number** field, specify your 12-digit serial number.
3. InfoExtended only: Run the InfoExec program to reconfigure your settings.

In the revised steps, the condition for each step is stated before the action. Users who are registered on the LAN can skip step 1, users who registered their software can skip step 2, and users who are not using InfoExtended can skip step 3.

The following step is introduced in a potentially confusing way:

Original

3. To specify the date parameter, click **New**.

Users might read the original step as optional, required, or conditional. The revisions show more specific phrasing for all three situations.

Revision: optional

3. Optional: Click **New** to specify the date parameter.

Revision: required

3. Click **New** to specify the date parameter.

Revision: conditional

3. If your date parameter is not defined, click **New** to specify it.

Take care to clearly identify conditional steps. Use “If” and state conditions early for steps that do not apply in all situations.

In sum

Task-oriented information focuses on the user's tasks and is presented from the user's perspective. Divide task information into tasks and subtasks. Provide steps that are clear, imperative, and grouped for usability. Do not clutter task topics with conceptual information. Get your users "on task" as quickly as possible.

This chapter provides guidelines to help you ensure that your topics are task oriented. Refer to the examples in the chapter for practical applications of these guidelines.

When you review technical information for task orientation, you can use the checklist on page 46 in two ways:

- ❑ As a reminder of what to look for, to ensure a thorough review
- ❑ As an evaluation tool, to determine the quality of the information

Based on the number and severity of items that you find, decide how the information rates on each guideline for this quality characteristic. You can then add your findings to "Quality checklist" on page 387, which covers all the quality characteristics.

Although the guidelines are intended to cover all areas for this quality characteristic, you might find additional items to add to the list for a guideline.

Guidelines for task orientation	**Items to look for**	**Quality rating**
Write for the intended audience.	• Audience is clearly defined. • Tasks are appropriate for the intended audience.	1 2 3 4 5
Present information from the user's point of view.	• Information is directed at the user. • Information denotes what the user does, as opposed to what the product does. • Contextual help reflects the user's position.	1 2 3 4 5
Indicate a practical reason for information.	• Details relate to tasks. • Conceptual information supports the task. • Only a necessary amount of conceptual information appears before a task.	1 2 3 4 5
Focus on real tasks, not product functions.	• Information focuses on real tasks, not artificial tasks. • Focus is on tasks, not features and interface elements.	1 2 3 4 5
Use headings that reveal the tasks.	• Headings reveal tasks. • Headings do not contain pseudo tasks.	1 2 3 4 5
Divide tasks into discrete subtasks	• Relationship of tasks to subtasks is clear. • Subtasks are discrete.	1 2 3 4 5
Provide clear, step-by-step instructions	• The first sentence of each step includes an imperative verb. • Steps are weighted appropriately. • Substeps and unordered lists are used appropriately. • Optional steps are clearly marked. • Conditional steps begin with the condition.	1 2 3 4 5

Note: The scale for the quality rating goes from very satisfied (1) to very dissatisfied (5).