Appendix B
Task Writing Guidelines

Introduction

Task statements are exemplars of the types of activities workers in an occupation may perform. Tasks are typically conceptualized as the smallest unit of activity with a meaningful outcome.

Cunningham (2000) states “Task writing requires considerable individual judgment. It is an art, and no two writers will produce exactly the same results. It is essential, however, that writers follow common guidelines in order to maintain some uniformity of task-statement structure.” To ensure consistency of structure and style across O*NET tasks, we implemented task writing procedures and guidelines.

This paper summarizes guidelines to assist in establishing uniformity of O*NET task statements. This paper presents information on the basic structure of O*NET task statements, and then describes elements of writing style for O*NET task statements.

Task Statement Structure:

Each task statement should follow the following basic format:

Action (Behavior) >

Object of the Action >

Purpose/Result, Enabler, and/or Context

Some task statements are simple, describing only the action and object of the action (with an implied or implicit purpose/result). Other task statements are more complex, including some or even all the component parts described above.

Generally, it is best not to write complex, compound task statements involving multiple action and purpose components. When task statements are too complex, consider writing multiple task statements, rather than a single comprehensive statement.

Each component part of the task statement structure is described briefly in the table below.

Component Part	Purpose	Description
Action (Behavior):	What action does the worker perform on, to or with the object?	The action or behavior is indicated by a verb, which introduces the task statement. The verb should be as specific and concrete as possible. The verb tense should assume a plural subject in the present tense. There may be one or more actions in a task statement.
Object of the Action:	What object is affected by the action or behavior? To whom or what is the action directed?	The object is the noun of the task statement, often with some modifying phrases. The object of the action will usually include Things, Data or People. There may be one or more objects in each task statement.
Purpose/Result:	What is the intended outcome of the behavior?	This part of the task statement may be implicit or explicit. When outcomes are explicit, they usually are introduced by a “to” statement.
Enabler:	An enabler is a phrase that describes how the action is supported so that the purpose or result may be achieved.	Enabling statements may include machines used, tools used, equipment used, sources of information used, methods used, knowledge and skill drawn upon, or the nature and specificity of the instructions followed. Enabler phrases typically are introduced with words such as using, referring to, relying on, following, based on, etc.
Context:	Context phrases describe the setting or conditions under which the task is performed.	Context phrases often appear at the end of a task statement, or immediately following the object of the action, and preceding the purpose/result. Context phrases are generally rare in task statements.

Task Writing Style:

In addition to the basic structure of task statements, some additional style guidelines have been implemented to achieve greater consistency among task statements. These guidelines are described below.

When writing task statements,

- Assume a plural subject (i.e., “Complete assignments” is correct, while “Completes assignments” is incorrect).

- Spell out all acronyms and abbreviations, and place the acronym or abbreviation in parentheses immediately following each occurrence.

- Do not use semicolons. A semicolon implies that there are two distinct tasks in one statement. Consider writing two task statements when a semicolon is needed, or use “and” or “or” to join two parts of a single task.

- With the exception of those used for acronyms or abbreviations, O*NET task statements should not include parentheses.

- Do not use “e.g.” or “i.e.” Instead, a “such as” statement may serve the same purpose.

- Do not use “and/or.” When “and/or” is appropriate, use “or.”

- Check the use of “and” when “or” may be more appropriate. Generally, “or” should be used more often than “and” in O*NET task statements. When “and” is used, job incumbents may believe they need to do every aspect of the task to endorse it. “Or” is more inclusive.

- When writing “such as,” or similar statements, use the word “and” to complete the example list. For instance, “Order supplies such as paper, pencils and erasers.”

- When writing “using,” or similar statements, use “or” to complete the example list. For example, “Order supplies such as paper, pencils and erasers, using telephone or computer.”

- Do not use forward slashes in task statements. Instead, substitute forward slashes with “and” or “or”.

- Do not use the phrase “in order to.” The word “to” is more concise and conveys the same meaning.

- Use hyphenated words sparingly, and check for proper and consistent hyphen use. Many words are spelled correctly with or without a hyphen. For instance, among the O*NET task statements, one will find “healthcare,” “health care,” and “health-care.” To achieve greater consistency, in such cases, opt not to use a hyphen. In this case, “health care” is the preferred spelling.

- Do not write task statements with time dependency. For instance, “do this, then do that.” When time dependency appears in a task statement, it implies that the statement is composed of two distinct tasks.

- Do not use “etc.” in task statements.

- Use the word “use” rather than “utilize.” This usage note was found:
  "Utilize", borrowed in the 19th century from the French: "utiliser", means 'make practical or effective use of.' Because it is a more formal word than "use" and is often used in contexts (as in business writing) where the ordinary verb "use" would be simpler and more direct, "utilize" may strike readers as pretentious jargon and should therefore be used sparingly.